Resum

Aquest projecte documenta el procés de disseny, millora i validació d'un sistema d'enregistrament de dades per ser implementat al CAT08e, el monoplaça elèctric que ha presentat l'ETSEIB Motorsport per competir a la Formula Student durant la temporada 2014/15.

Aquesta memòria descriu tot el procés de disseny i fabricació, tant del hardware com del software. S'explicarà des de la selecció i funcionament dels components fins a l'estructura del programa principal, així com el desenvolupament d'una interfície gràfica per tal de tractar i processar totes les dades guardades. Com a introducció s'analitzen els antecedents i les diferents possibles solucions per tal d'escollir una opció vàlida i fiable que satisfaci totes les especificacions. Això ha suposat que per primer cop a l'equip s'utilitzés un microcontrolador de 32 bits, amb totes les dificultats que suposa aquest important canvi.

Cal destacar que durant la temporada anterior ja es va desenvolupar un mòdul d'adquisició de dades. Davant la necessitat de guardar dades a una major freqüència, els membres de la secció d'electrònica vam decidir desenvolupar una nova solució que pogués cobrir tots els requeriments.

Finalment, aquest document pretén dotar de nou coneixement a futurs equips d'ETSEIB Motorsport, de tal forma que se'n pugui treure el màxim profit per millorar el disseny electrònic dels vehicles futurs.

Sumari

1. Glossari

BMS (*Battery Management System*): conjunt de plaques que monitoritzen i controlen l'estat de les cel·les de les bateries d'alt voltatge del cotxe.

BOM (*Bill of Materials*): llista de materials i components necessaris per fabricar un producte.

Bus: sistema digital de comunicació que transfereix dades entre components electrònics.

CAN (*Controller Area Network*): protocol de comunicacions entre ECUs molt emprat en l'àmbit de l'automoció per la seva robustesa.

Datasheet: document que resumeix totes les característiques tècniques d'un producte.

ECU (*Electronic Control Unit*): sistema electrònic integrat que controla un o més sistemes electrònics.

Esquemàtic (*Schematic*): representació gràfica d'un circuit elèctric on s'hi representen els components i connexions usant símbols estandarditzats.

Fitxer mat: fitxer propi de l'entorn de programació MATLAB que permet emmagatzemar valors de variables.

Fitxer txt: fitxer que emmagatzema informació estructurada com una seqüència de línies de text.

Ground: part de l'alimentació d'un circuit elèctric que es troba a 0V de potencial.

GUI (*Grafical User Interface*): interfície gràfica que permet a l'usuari interaccionar amb una sèrie de funcions i tasques en un entorn visual.

GUIDE (*GUI Development Environment*): entorn de desenvolupament per a la programació d'interfícies gràfiques mitjançant MATLAB.

Hardware: vessant de l'electrònica que profunditza sobre "tot allò que es pot tocar": disseny i fabricació de PCBs, selecció de components, etc.

Inversor: component del tren de potència del cotxe encarregat de convertir en tensió alterna la tensió contínua proporcionada per les bateries, actuant com a controlador del motor del cotxe.

LED (*Light-Emitting Diode*): component electrònic consistent en un díode emissor de llum.

Microchip: fabricant americà de microcontroladors, memòries i semiconductors analògics.

Microcontrolador: circuit integrat programable, capaç d'executar funcions gravades a la seva memòria.

PCB (*Printed Circuit Board*): suport mecànic que connecta elèctricament un conjunt de components electrònics.

PIC (*Programmable Interface Controller*): família de microcontroladors d'arquitectura Harvard fabricats per la companyia Microchip Technology Inc.

Ports GPIO (*General-Purpose Input/Output*): pin genèric en un microcontrolador el comportament del qual es pot controlar.

PTC (*Positive Temperature Coeficient*): materials que experimenten un increment de la seva resistència elèctrica quan augmenta la seva temperatura.

SD *Card* (*Secure Digital Card*): targeta de memòria no-volàtil molt comú en telèfons mòbils, càmeres i ordinadors portàtils.

Software: vessant intangible de l'electrònica, com a tasca principal dissenya el codi amb les instruccions que executen els microcontroladors.

SPI (*Serial Peripheral Interface Bus*): protocol de comunicació en sèrie molt comú en distàncies de comunicació curtes. Habitual en sistemes encastats, control de sensors i targetes SD.

Telemetria: tecnologia que permet el mesurament a distància i la comunicació d'informació. El terme deriva de les arrels gregues *tele* ("a distància") i *metron* ("mesurar").

TQ (*Time Quanta*): en el protocol CAN es tracta de cadascun dels segments mínims de temps en que es poden dividir les diferents fases d'enviament de cadascun dels bits. És un múltiple del senyal de rellotge de l'oscil·lador.

Transceptor de CAN: component electrònic que fa de receptor i transmissor de missatges amb el protocol bus CAN.

Through-Hole: forats passants de les PCB que s'utilitzen pel muntatge i soldadura dels components electrònics.

USB (Universal Serial Bus): bus sèrie industrial que defineix els cables, connectors i protocols utilitzats en un bus per connectar, alimentar i comunicar dispositius electrònics.

2. Índex de Figures

3. Índex de Taules

4. Prefaci

4.2. Origen del projecte

El projecte s'emmarca dins l'equip ETSEIB Motorsport, un grup d'estudiants de l'Escola Tècnica Superior d'Enginyeria Industrial de Barcelona que dissenyen i fabriquen un monoplaça per competir a la Formula Student, on hi participen altres universitats d'arreu del món. Cada universitat presenta el seu cotxe i competeix en diferents tipus de proves on un jurat, format per professionals de l'automoció, determina la puntuació de cada equip. Les proves es divideixen en dos grans blocs: les proves estàtiques, on s'avalua la capacitat de disseny i màrqueting de l'equip; i les proves dinàmiques, en les que membres de l'equip piloten els vehicles per aconseguir el millor temps en diferents modalitats.

L'equip va formar-se l'any 2007 i des de llavors ha presentat un monoplaça cada temporada. La temporada 2014/15, l'ETSEIB Motorsport participarà en dues competicions amb el CAT08e, el 8è prototip (el 4t elèctric) de l'equip. El projecte que ens ocupa s'inclou precisament en el disseny electrònic d'aquest model.

4.3. Motivació

En el món de l'automoció, i especialment de competició, és imprescindible disposar de la màxima informació del cotxe. És per això que un cotxe de Fórmula 1 pot anar equipat amb gairebé 200 sensors que recullen informació de qualsevol part del cotxe. Recollir i emmagatzemar aquestes dades és clau per després poder analitzar-les per treure'n conclusions i així millorar i conèixer amb profunditat el comportament del vehicle. Hi ha dades que, per les seves característiques, requereixen ser guardades a freqüències molt elevades, fet que fa augmentar la complexitat d'un mòdul d'adquisició de dades.

A part de la motivació que suposava aquest repte, el projecte és una oportunitat idònia pels estudiants. En ells recau la responsabilitat de prendre les decisions oportunes per aconseguir desenvolupar un ambiciós projecte en un termini de temps molt ajustat. Tot això es fa possible gràcies als recursos econòmics que aporten l'escola i els patrocinadors, de l'assessorament del professorat i de l'apassionada dedicació de cadascun dels membres de l'equip.

5. Introducció

5.1. Objectius del projecte

L'objectiu del projecte és dissenyar i construir un mòdul d'enregistrament de dades que millori les prestacions dels sistemes utilitzats anteriorment a l'equip. La finalitat és enregistrar totes les dades a una freqüència adequada per al posterior anàlisi.

Aquesta centraleta també pretén unir el sistema d'enregistrament de dades amb el sistema de telemetria, desenvolupat la temporada passada per un membre de l'equip. Amb això es vol reduir el volum, el pes i el cablejat, així com evitar les duplicitats en l'enviament de dades.

La finalitat de la centraleta és oferir una solució satisfactòria pel CAT08e. Tot i així, aquesta centraleta podria adaptar-se a futurs prototips de l'equip si els membres així ho decideixen. En tot cas, s'espera que el coneixement adquirit serveixi a futurs membres per dissenyar un nou mòdul o millorar-ne d'altres.

5.2. Abast del projecte

El projecte abasta tant sols el sistema d'enregistrament de dades, excloent-ne tot el sistema de telemetria, tot i coexistir a la mateixa placa de circuit imprès. Al ser un projecte d'enginyeria electrònica, aquest engloba totes les fases del procés. Així doncs, les tasques o fases a desenvolupar seran:

- Anàlisi i tria del sistema i protocol d'enregistrament més adequat.
- Tria del microcontrolador capaç de complir amb les especificacions.
- Hardware: cerca i selecció dels components, disseny dels esquemes electrònics amb les connexions, disseny de la placa de circuit imprès, muntatge i validació.
- Software: recerca del programari necessari, paquet de llibreries, programació del codi del microcontrolador i validació del programa.
- Desenvolupament de la interfície gràfica per processar les dades.
- Test i validació del funcionament de la centraleta al CAT08e.

Tot el procés d'anàlisi de dades queda fora de l'abast d'aquest projecte, ja que és responsabilitat de cada departament (dinàmica, tren de potència, electrònica, etc) analitzar-les i prendre les decisions oportunes.

6. Anàlisi d'antecedents i viabilitat

6.1. Antecedents a l'ETSEIB Motorsport

Abans de prendre cap decisió, és convenient fer un anàlisi dels antecedents. S'han contemplat aquelles solucions adoptades per l'equip en temporades anteriors, veient-ne les avantatges i inconvenients que van suposar.

6.1.1. Mòdul comercial

Els primers prototips elèctrics de l'equip i la majoria d'universitats fan servir un mòdul comercial, que és probablement la solució amb millors prestacions. Tot i així, té el principal inconvenient del cost, ja que està dissenyat per a equips professionals de competició, la qual cosa fa que sigui una solució sobredimensionada amb un consum superior a les altres opcions. També cal tenir en compte que la Formula Student és una competició d'enginyeria on es premia el grau d'intervenció dels estudiants en el disseny dels components, i una solució comercial no aporta valor afegit al disseny.

Figura 6.1. Mòdul comercial d'adquisició de dades. Font: dassa.es

6.1.2. Mòdul semi-comercial

Equips anteriors de l'ETSEIB Motorsport, van utilitzar un mòdul hardware comercial de la marca Arduino. Amb les llibreries i possibilitats que ofereix, es va desenvolupar un software propi per tal de guardar totes les dades en una targeta SD. Aquesta solució és més interessant de cara a la competició, però segueix sent sobredimensionada i poc flexible en quant a volum i pes.

6.1.3. Mòdul propi

La temporada passada (2013-14) es va prendre la decisió d'elaborar un mòdul *self-made*, completament dissenyat i muntat per l'equip. Es va dissenyar la placa de circuit imprès i tot el software. Això implicava escollir el microcontrolador i el sistema d'enregistrament de dades. Es va escollir el mateix microcontrolador que feien servir per les altres centraletes per guardar les dades en un fitxer de text en una targeta SD. El principal obstacle va ser que el microcontrolador no era prou potent per executar totes les funcions a la velocitat que era necessària. Això va implicar la creació d'unes llibreries pròpies i molts problemes a l'hora de desenvolupar el software. També es va haver de duplicar tot el mòdul perquè fos capaç de guardar totes les dades i, tot i així, no va acabar de complir els requeriments.

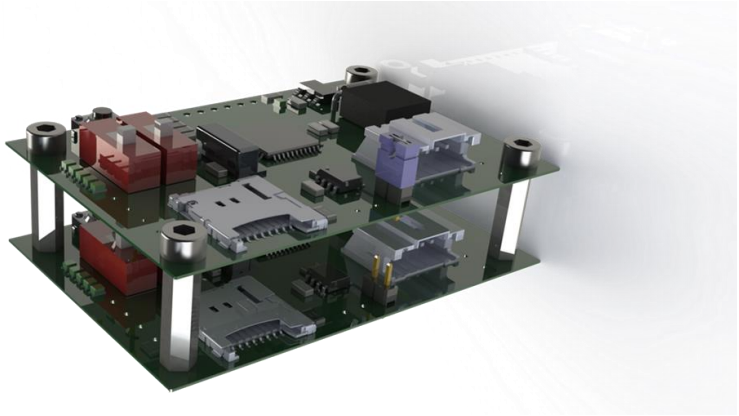


Figura 6.2. Mòdul propi d'adquisició de dades. Font: [3].

Aquesta solució, junt amb tot el coneixement adquirit en el seu disseny, han sigut els predecessors d'aquest projecte, que pretén ser la seva evolució lògica. Tota la informació queda documentada al document MAGRIÑÁ CLEMENTE, D. *Mòdul d'adquisició de dades per a l'equip ETSEIB-Motorsport Barcelona de la Formula Student*. Barcelona, ETSEIB, 2014. De les conclusions d'aquest treball, se n'extreuen els següents aspectes clau a millorar i a tenir en compte:

- Utilitzar un microcontrolador més potent (per exemple, de la família PIC32), que sigui capaç d'executar totes les funcions a una freqüència més elevada.
- Transferir el màxim de responsabilitats a la interfície gràfica, alliberant de certes operacions al microprocessador. Això fa referència, per exemple, al tractament o condicionament de les variables.
- Estructurar el codi en capes, de tal forma que sigui més comprensible i interpretable de cara a futurs membres o qualsevol programador que vulgui fer-ne modificacions.

6.2. Viabilitat

6.2.1. Viabilitat tècnica

Una centraleta que tingui aquestes prestacions és totalment viable des del punt de vista tècnic. Si bé és cert que s'haurà de buscar un nou processador, l'experiència de disseny d'altres anys representa un bon punt de partida. En quant al hardware de l'entorn del processador, aquest serà molt semblant a les centraletes anteriors. Per un altre costat, tots els components electrònics necessaris es troben a l'abast de l'equip gràcies, en part, al suport dels múltiples patrocinadors.

Pel que fa al software caldrà dedicar-hi més hores, ja que no hi ha antecedents amb nous microprocessadors. Tot i així, tots ells estan molt ben documentats i hi ha un gran ventall de llibreries i informació per desenvolupar cadascuna de les funcions. També caldrà fer un programa ben estructurat que permeti fer-lo entenedor i modificable per a futurs membres.

6.2.2. Viabilitat ambiental

La viabilitat ambiental no suposa un aspecte limitant en el desenvolupament d'aquest projecte, ja que només se'n fabricarà un prototip. Hi ha un seguit de directrius internacionals (com per exemple, la RoHS) que restringeixen l'ús de certs materials perillosos en la fabricació de components electrònics.

Al punt *14. Impacte mediambiental* d'aquest document s'exposa un estudi de l'impacte ambiental més detallat.

6.2.3. Viabilitat econòmica

Com ja s'ha comentat, només es fabrica un prototip del vehicle i per tant, el nombre de centraletes d'enregistrament de dades també queda limitat a un parell o tres com a màxim. A més a més, l'equip compta amb molts patrocinadors que cobreixen els costos de gairebé tot el material i els processos de la secció d'electrònica. Així doncs, queda cobert el cost dels components, la fabricació de les PCBs, el cablejat i els connectors.

Al punt *13. Pressupost econòmic* d'aquest document, es pot trobar un estudi econòmic més desenvolupat i extens.

7. Especificacions

7.1. Condicions generals

Un sistema d'enregistrament de dades pot presentar moltes propostes de solució. Per tant, cal acotar el màxim possible el projecte. És per això que s'han definit exactament les funcions que ha de realitzar la centraleta:

- Recepció dels missatges provinents de la centraleta principal que contenen totes les dades. S'utilitza el protocol de comunicació bus CAN, el qual es desenvolupa més extensament al punt *9.3 Àrea de comunicacions CAN*.
- Creació d'un fitxer de text a la unitat de memòria.
- Enregistrament continu de totes les dades rebudes a la freqüència desitjada.

Per un altre costat, cal comentar que la centraleta estarà alimentada per la bateria de baix voltatge a 12V, per tant caldrà tota una àrea d'alimentació per adequar el voltatge dels diferents components.

7.2. Entorn del mòdul

La centraleta principal del cotxe és l'encarregada d'enviar totes les dades al mòdul mostrejades cada cert temps. La centraleta recull les dades dels diferents sensors repartits pel cotxe. Algunes d'aquestes dades són tractades prèviament per altres centraletes. Les centraletes i sensors, que envien les dades a la centraleta principal, són:

- **Front ECU.** La centraleta frontal del cotxe recull la informació dels potenciòmetres de direcció, suspensions, accelerador i el sensor de pressió de frens de davant.
- **Rear ECU**. La centraleta posterior, a part de realitzar certes funcions, també recull i envia dades dels potenciòmetres de suspensions i del sensor de pressió dels frens del darrera.
- **LowVoltage Board.** És la placa que s'encarrega de l'alimentació de baix voltatge. Aquesta transmet les dades d'intensitat dels circuits de baix voltatge i l'estat de les bateries de baix voltatge.
- **Sensor Òptic.** Obté dades sobre les velocitats i la distància recorreguda pel cotxe.
- **Battery Management System (BMS)**. És el sistema que regula el funcionament de les cel·les que conformen la bateria d'alt voltatge. Envia dades a la centraleta principal de l'estat de totes les cel·les. La centraleta envia al mòdul d'adquisició de

dades només el número i temperatura de la cel·la a temperatura màxima i el número i voltatge de les 5 cel·les amb el voltatge més baix.

- **Inversor.** S'encarrega de convertir el corrent continu a altern per proporcionar potència als motors. Envien informació sobre les tensions i intensitats del bus de continua i altres paràmetres. També envien la freqüència a la que treballa el motor.
- **Altres**. Hi ha una sèrie de sensors que són llegits directament per la centraleta principal, ja sigui per la seva proximitat o perquè el sensor és intern a la centraleta. Aquests són els sensors de temperatura (motor, inversor i transmissió) i les 3 components de l'acceleròmetre que porta incorporat la centraleta principal.

La *Figura 7.1.* pretén il·lustrar de forma gràfica i esquemàtica la transmissió de totes aquestes dades al mòdul d'enregistrament.

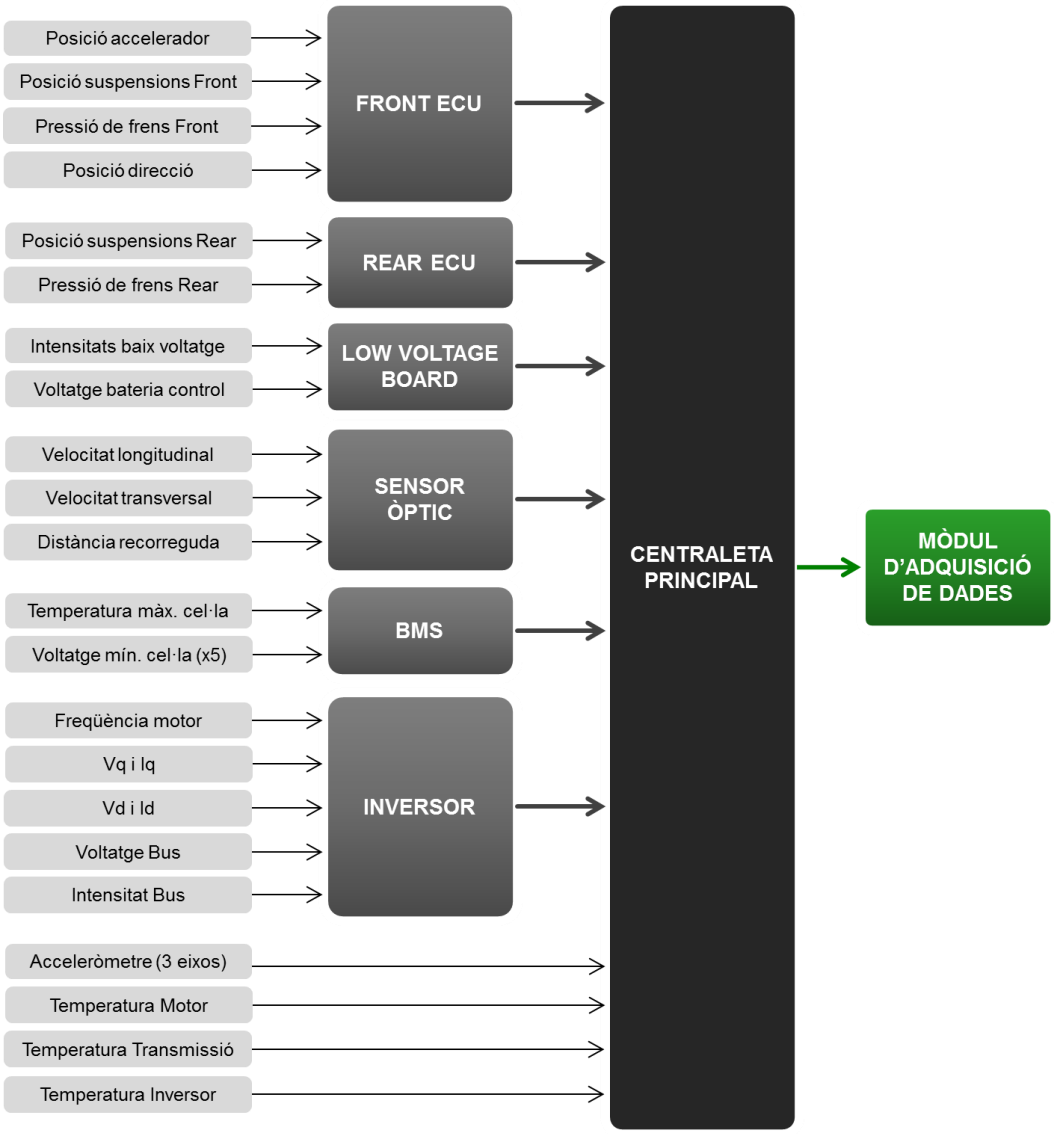


Figura 7.1. Esquema general d'obtenció de dades. Font: pròpia.

7.3. Dades a recollir

El total de dades diferents a recollir és de 47 magnituds. És molt important fer un anàlisi de les freqüències desitjades per a cadascuna de les dades. Les freqüències adequades per a cada tipus de dada són:

Tipus de dada	Freqüència adequada
Temperatures	1 Hz
Freqüència motor	20 Hz
Posició accelerador	20 Hz
Posició suspensions	100 – 200 Hz
Velocitats	20 Hz
Acceleròmetres	20 Hz
Pressió de frens	10 Hz
Direcció	10 – 50 Hz
Voltatges	5 Hz
Intensitats	5 Hz

Taula 7.1. Freqüències adequades per a cada tipus de dada. Font: pròpia.

S'observa que la freqüència requerida més alta és la de la posició de les suspensions. A nivell d'alta competició s'utilitzen freqüències de 200 Hz o superiors. S'ha considerat que per un cotxe de Formula Student, i en particular el nostre, serà suficient guardar-ho a una freqüència de 100 Hz. Si aquesta freqüència resulta ser un limitant a l'hora de fer els anàlisis de dades, es podria estudiar la possibilitat d'augmentar-la.

Per tal de simplificar el programa i el posterior tractament de dades, s'ha decidit guardar totes les dades a aquesta freqüència de **100 Hz**. D'aquesta forma s'aconsegueix satisfer tots els requeriments pel que fa a la freqüència de mostreig.

També interessa tractar les dades per tal que es puguin enviar amb el mínim de missatges possibles. Cada missatge agrupa les dades en bytes, per tant s'intentarà enviar els missatges en un sol byte (8 bits). En alguns casos és necessari més d'un byte per guardar la dada amb la resolució adequada. Per tal d'optimitzar-ho, se'ls hi aplicarà una correcció (un guany i un desfasament) per tal d'aconseguir que aquest valor estigui comprés entre 0 i 255, que són els valors que es poden enviar en un byte (2^8 = 256). D'aquesta forma s'aconseguirà una certa resolució, que en cas de ser inferior a la desitjada, s'haurà d'utilitzar més d'un byte.

A part d'això, la centraleta també enviarà el **temps**. Com que envia totes les dades cada 100 Hz, això correspon a un bloc de dades cada 10 ms. Així doncs, aquest valor serà un comptador de 0 a 200, que s'incrementarà cada cop que enviï un nou paquet de dades.

També es destinarà un byte per enviar totes aquelles dades binàries, utilitzant un bit per a cadascuna d'elles. Com que la majoria d'aquestes seran indicadors de fallada (*failures*), s'anomenarà així. Tot i així, també n'hi haurà d'altres tipus, com l'estat del DRS (*Drag Reduction System*).

Finalment, i per acabar d'ocupar tots els missatges, es destinaran dos bytes a variables auxiliars, per si en algun moment o prova específica és necessari registrar alguna dada addicional.

La *Taula 7.2.* conté una mostra de les dades que ha d'enregistrar el mòdul amb les seves característiques. També s'indica, per a cadascuna d'elles, la correcció (guany i desfasament) que se li aplicarà i la resolució que li correspon. A l'*Annex A* es troba la taula sencera de totes les dades amb les seves transformacions.

Codi	Dada	Unitats	Rang REAL	Desfas.	Guany	Resolució	Núm. bits	Núm. bytes
TIME	Temps	ms	[0 ; 2000]	0	0,1	10	8	1
FAIL	Failures*	-	[0 ; 255]	0	1	-	8	1
SUFR	Suspensió *Front Right*	mm	[-20 ; 20]	20	10	0,1	9	2
AC1X	Acceleròmetre – Eix X	g	[-2 ; 2]	2	50	0,02	8	1
						...		
VLON	Velocitat Longitudinal	m/s	[0 ; 35]	0	100	0,01	12	2
VTRA	Velocitat Transversal	m/s	[0 ; 25]	0	10	0,1	8	1
DIST	Distància Recorreguda	m	[0 ; 25500]	0	2	0,5	16	2

Taula 7.2. Característiques i correcció de les dades a enregistrar. Font: pròpia.

Això suposa un total de **49 dades** a guardar en un total de **56 bytes**. Tal com s'exposarà més endavant, això representa un total de 7 missatges de CAN. D'aquestes 49 dades, n'hi ha 42 que necessitaran només 1 byte i unes altres 7 dades que en necessitaran 2.

Aquest tractament de les dades, i amb el fi d'alliberar de responsabilitats al microcontrolador, es realitza a la centraleta principal i se li aplica l'invers a la interfície de l'ordinador.

7.4. Sistema d'enregistrament

Un altre aspecte que és necessari definir és el sistema d'enregistrament de les dades. A la *Taula 7.3.* es fa una comparació de les diferents opcions:

	Micro SD		**USB**	
	Mode SD 4-bit	Mode SPI	2.0 (High Speed)	3.0 (Super Speed)
Pins totals	8		4	
Pins utilitzats	8	5	4	4
Pins de dades	4 (DATA 1, 2, 3 i 4)	2 (DATA IN i OUT)	2 (D+ i D-)	2 (D+ i D-)
Protocol de comunicació amb el microcontrolador	SD paral·lel/síncron	SPI sèrie/síncron	USB diferencial/asíncron (half-dúplex)	USB diferencial/asíncron (full-dúplex)
Velocitat màxima d'escriptura	25 MB/s	12,5 MB/s	40 MB/s	160 MB/s
Velocitat màxima de lectura	25 MB/s	12,5 MB/s	40 MB/s	240 MB/s
Memòria màxima	4 GB		1024 GB (1 TB)	
Compatibilitat PIC18	Sense referents	Sí	No	No
Compatibilitat PIC32	Sense referents	Sí	Sí	No
Alimentació	3,3 V		5 V	
Durabilitat	10.000 cicles		1.500 cicles	

Taula 7.3. Comparació dels sistemes d'enregistrament. Font: pròpia.

En el disseny del prototip anterior, el CAT07e, la unitat de memòria (*flash memory*) utilitzada era una targeta microSD. El microcontrolador era un PIC18LF4580 i feia servir un protocol de comunicació SPI (*Serial Peripheral Interface*) amb la targeta microSD. Era necessari incrementar la velocitat d'enregistrament de dades i això passava per augmentar la freqüència del microcontrolador i buscar una comunicació més ràpida amb la unitat de memòria.

La primera opció que es va contemplar va ser la de mantenir el mateix tipus de unitat de memòria i intentar fer servir el mode SD 4-bit que feia servir 4 pins de dades enlloc de 2, la qual cosa permetia duplicar la velocitat d'escriptura. L'inconvenient principal d'aquesta opció era que no hi havia cap referent ni documentació de cap projecte que comuniqués un microcontrolador de la companyia Microchip amb una targeta microSD utilitzant el mode SD 4-bit. Aquest fet va fer descartar aquesta opció.

Seguidament es van analitzar altres tipus d'unitats de memòria. Actualment, l'opció més ràpida són les memòries USB. La interfície USB 2.0 pot assolir velocitats d'escriptura de 40 MB/s i la USB 3.0 de fins a 160 MB/s. D'aquesta forma es triplicava la velocitat màxima de les targetes SD, la qual cosa donava força a aquesta opció. La interfície USB 3.0 no té compatibilitat amb els microcontroladors de Microchip, però sí la USB 2.0, que és compatible amb els PIC32. Aquests microcontroladors disposen d'un paquet de llibreries (veure punt ***¡Error! No se encuentra el origen de la referencia. ¡Error! No se encuentra el origen de la referencia.***) amb molts exemples de comunicació USB entre una unitat de memòria i el microcontrolador. Això suposa un punt de partida molt favorable i és per això que s'ha escollit aquesta solució. També aporta uns coneixements nous a l'equip, que si més no ampliaran el bagatge i l'experiència d'aquest per a futurs dissenys basats en microcontroladors de 32 bits.

7.5. Característiques tècniques

Després de prendre decisions sobre alguns aspectes tècnics, se'n poden extreure una sèrie de característiques tècniques que definiran i condicionaran el disseny del mòdul. Aquestes especificacions queden resumides a la següent taula:

Especificació	Valor	Observacions
Alimentació	12 V	L'alimentació externa de la placa serà de 12 V, proporcionada per la bateria de baix voltatge.
Consum	500 mA	El dimensionament de la bateria de baix voltatge limita el consum a 500 mA.
Comunicació	Bus CAN	Comunicació utilitzada per la centraleta principal i per la resta de centraletes.
Sistema d'enregistrament	USB 2.0	Es guardaran les dades en una memòria *flash* amb comunicació USB 2.0.
Temps de funcionament	40 min	Haurà de ser capaç d'estar en funcionament continu durant 40 minuts (duració de la prova més llarga).
Microcontrolador	PIC32	Compatible amb la comunicació USB i CAN. Freqüència màxima de funcionament de 80MHz. Alimentació a 3,3V.
Nombre de dades a guardar	49	Es guardaran un total de 49 dades a cada mostreig. (justificat al punt *7.3 Dades a recollir*)
Freqüència de guardat	100 Hz	Totes les dades es guardaran a 100 Hz, cada 10ms. (justificat al punt *7.3 Dades a recollir*)
Tipus de fitxer	TXT	Facilita i agilitza l'escriptura i posterior tractament de les dades.

Taula 7.4. Especificacions tècniques de la centraleta. Font: pròpia.

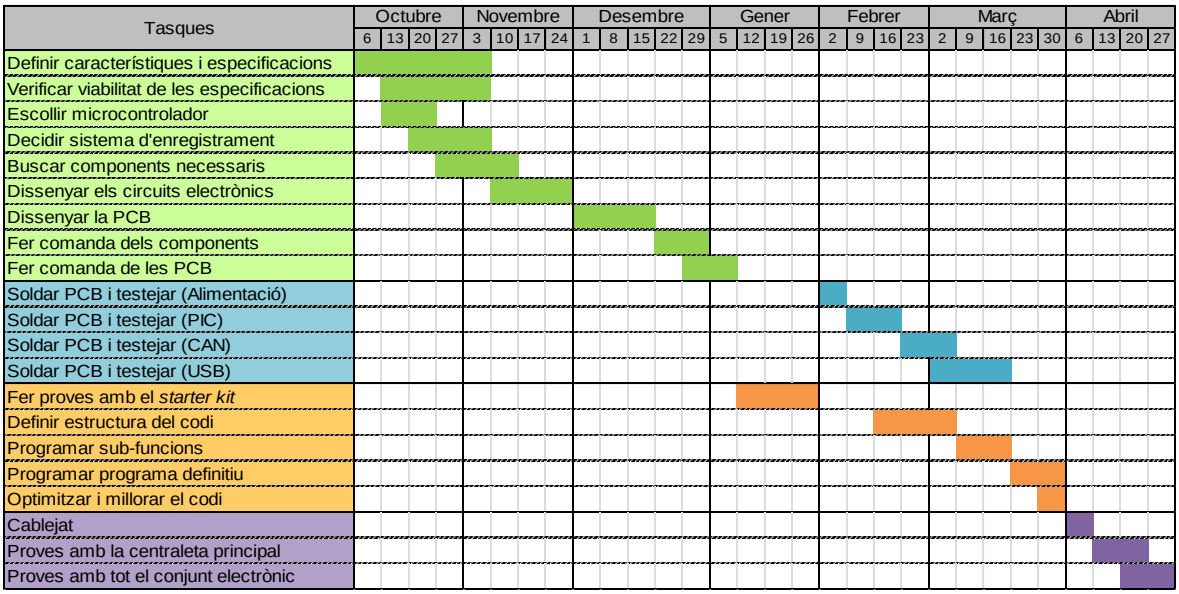
8. Planificació

El projecte té una durada aproximada de 7 mesos, coincidint amb la temporada de l'equip. S'inicia a l'octubre del 2014 i finalitza a l'abril del 2015, tenint en compte que la dedicació és parcial. La dedicació mitjana és de 4 hores al dia, amb períodes més intensos que altres.

Es poden distingir una sèrie de fases:

- Fase de disseny: és la fase inicial del projecte. Aquesta etapa s'inicia amb la recerca i recopilació d'informació i és on es prenen la major part de les decisions. Les tasques pròpies d'aquesta fase són: selecció dels components, disseny dels circuits electrònics, disseny de la PCB i comandes als proveïdors.
- Fase de muntatge: aquesta fase s'inicia en el moment en que arriben les PCB i els components. Aleshores s'inicia una fase de soldadura i proves amb els subsistemes de la placa.
- Fase de software: Un cop s'han soldat els components, s'entra en l'etapa de desenvolupar el programa. Aquesta etapa consisteix en anar programant i provant les funcions del programa fins a assolir el programa complet.
- Fase de test: Amb la centraleta enllestida, s'inicia l'etapa de proves amb totes les altres centraletes. Es comprova que tot funciona. També és important tenir més d'un prototip funcional per si algun d'ells es malmet.

La *Taula 8.1.* mostra un diagrama de Gantt on hi apareixen totes les tasques i fases del projecte amb la seva previsió de dates d'inici i fi.



Taula 8.1. Diagrama de Gantt amb les tasques del projecte. Font: pròpia.

9. Desenvolupament del hardware

El hardware de la centraleta es composarà d'una placa de dues capes, amb unes dimensions de 43x87 mm. La placa s'encarregarà, tal com ja s'ha comentat, tant del sistema d'enregistrament de dades com de la telemetria. Aquest document només explica les característiques del sistema d'enregistrament, tot i que molts elements són compartits.

La placa s'allotjarà dins d'una caixa d'alumini estanca, que actuarà com a gàbia de Faraday aïllant el sistema electrònic de possibles interferències.

El disseny de la placa estarà dividit en diverses zones segons les funcions que s'han de desenvolupar. A la *Figura 9.1.* es representa la placa sencera amb les seves zones.

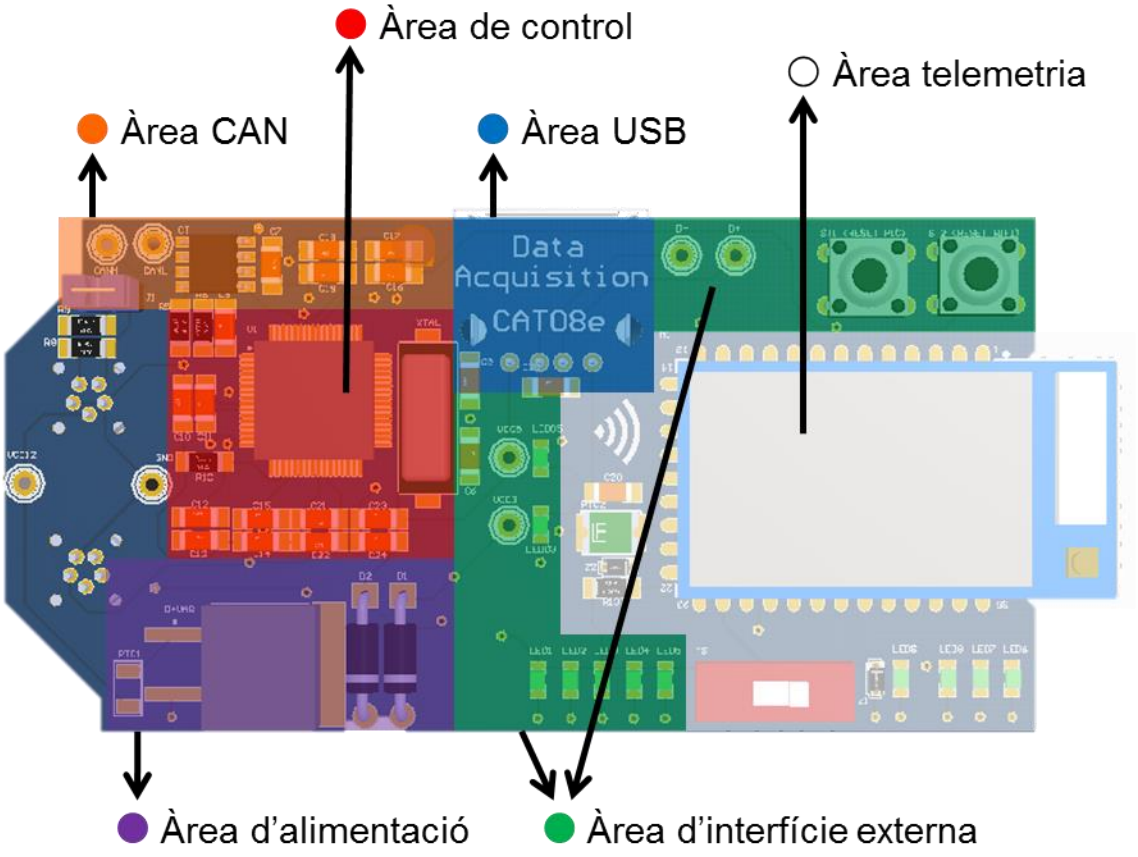


Figura 9.1. Distribució de les àrees de la PCB. Font: pròpia.

A continuació es farà una descripció detallada de cadascuna d'aquestes àrees i dels components que la conformen. Cal recordar que l'àrea de telemetria no entra dins l'abast d'aquest projecte i no s'explicarà.

9.1. Àrea d'alimentació

En primer lloc, cal dissenyar l'àrea d'alimentació. Aquest subsistema ha de convertir la tensió de 12 V de contínua que li arriba, a les tensions adequades de funcionament dels components. En aquest cas, haurà de convertir-los a 3,3 V per alimentar el microcontrolador i a 5 V per alimentar la memòria *flash*.

Aquesta àrea també haurà d'aïllar tota la resta de la placa de possibles irregularitats en l'alimentació, com per exemple pujades i baixades de tensió.

A la *Figura 9.2.* es mostra el circuit electrònic de l'àrea d'alimentació:

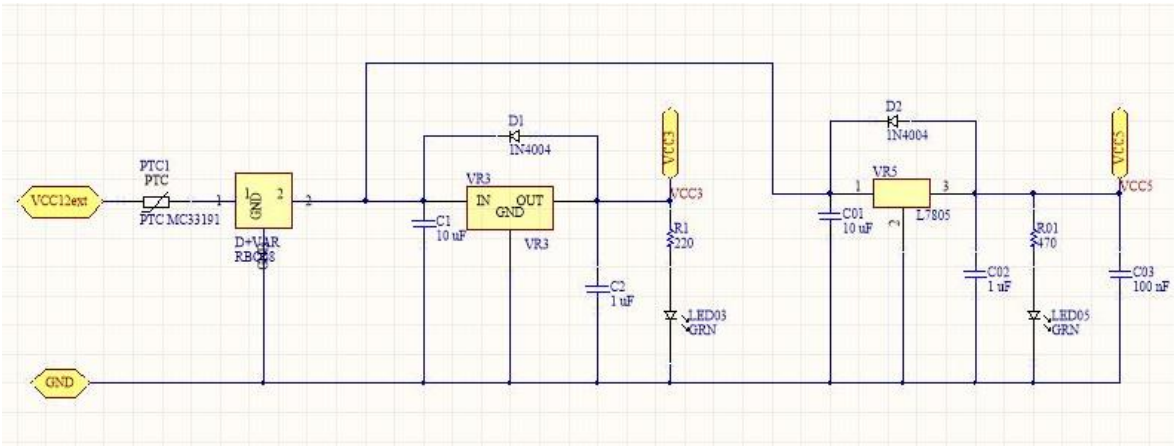


Figura 9.2. Esquemàtic de l'àrea d'alimentació. Font: pròpia.

Els components que conformen aquesta àrea, són:

- **Resistència PTC** *[9]* (PTC1): és un fusible auto-resetejable. Aquest component protegeix el circuit de sobrecorrents, ja que, quan circula més corrent del compte, s'escalfen i augmenten el seu valor òhmic fins a tallar el corrent. Quan tornen a una temperatura normal de treball tornen a tenir el valor resistiu nominal.

- **Díode + Varistor** *[10]* (D+VAR): és un altre element de protecció format per diversos díodes, els quals realitzen diverses funcions. El primer díode protegeix el circuit davant la possibilitat de tenir una alimentació amb la polarització invertida. Els altres dos components són díodes Zener que protegeixen davant de pics de tensió positius i negatius. D'aquesta forma, el conjunt dels 3 components protegeixen el circuit contra variacions de tensió excessius. A la *Figura 9.3.* es mostra el diagrama funcional d'aquest dispositiu.

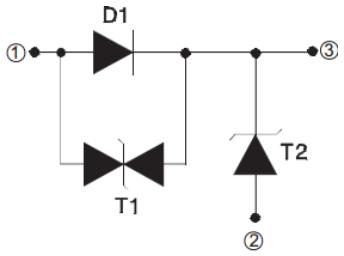


Figura 9.3. Diagrama funcional RBO08. Font: Datasheet RBO08 [10].

- **Regulador de tensió de 3,3 V** *[13]* (VR3): és el component encarregat de convertir la tensió d'alimentació (12 V) a la tensió adequada per a altres components. Tant el microcontrolador com el mòdul WiFi s'alimenten a una tensió de 3,3 V, de tal forma que és necessari un regulador d'aquest tipus. El component que s'ha escollit admet una tensió d'entrada d'entre 5,3 i 25 V, i un corrent màxim de 500 mA. Aquestes dues característiques compleixen amb les especificacions del mòdul..

- **Regulador de tensió de 5 V** *[11]* (VR5): el funcionament és igual que la del regulador de 3,3 V però el voltatge de sortida és de 5 V. La memòria *flash* USB ha d'estar alimentada a aquesta tensió i per aquesta raó és necessari aquest component. El component que s'ha escollit admet una tensió d'entrada d'entre 7,5 i 35 V, i un corrent màxim d'1 A. També compleix les especificacions.

Cadascun dels reguladors va acompanyat d'una sèrie de components. Hi ha un condensador a la entrada i sortida de cada regulador per estabilitzar el voltatge d'entrada i sortida. Han estat dimensionats segons les indicacions dels *datasheets*. També s'hi instal·la un díode que assegura que la tensió de la sortida no serà superior a la d'entrada.

Cal comentar que, un cop vist el resultat, s'ha valorat l'opció de substituir els reguladors de tensió per convertidors CC/CC. La principal avantatge d'aquests components és que són molt més eficients que els reguladors i, per tant, no s'escalfen tant. Aquesta millora en l'eficiència reduiria el consum de la centraleta. El principal inconvenient és que normalment són més grans que els reguladors utilitzats. De totes formes, es valora de forma positiva que, de cara a futurs dissenys, s'emprin convertidors CC/CC enlloc de reguladors lineals.

9.2. Àrea de control

L'àrea de control està formada bàsicament pel microcontrolador. Aquest és l'element clau de la centraleta. És l'encarregat de rebre, processar i gestionar les dades per tal que tota la resta compleixi les seves funcions. Tant és així, que requereix anar acompanyat d'una sèrie de components electrònics que garanteixin unes bones condicions de funcionament.

En aquest punt, s'ha de realitzar l'elecció de quin serà el microcontrolador. Com s'ha pogut intuir en altres punts del treball, s'optarà per un microcontrolador de l'empresa Microchip i concretament de la família dels **PIC32** (32-bit). El motiu d'aquesta elecció és bàsicament per assegurar complir el primer objectiu plantejat al punt *6.1.3 Mòdul propi*, on es destaca la importància de fer servir un microcontrolador que sigui capaç de treballar a una freqüència més elevada. El PIC18 utilitzat fins llavors podia treballar a una freqüència màxima de 40 MHz, mentre que alguns dels PIC32 poden treballar fins a 80 MHz.

Dins de la família dels PIC32 hi ha més 100 productes diferents i és per això que cal definir molt bé els requeriments de funcionament:

- Tensió de funcionament a 3,3 V. Com s'ha comentat abans, la tensió de funcionament de la placa, excepte per l'àrea de comunicació USB, serà de 3,3 V.
- Mòdul CAN. Necessari per llegir i enviar missatges de CAN a la xarxa. Ha de tenir com a mínim un mòdul de CAN.
- Mòdul UART. Necessari per comunicar-se amb el mòdul WiFi.
- Mòdul USB. Necessari per establir la comunicació amb la unitat de memòria USB.

Amb aquests requeriments mínims i a fi d'assegurar unes bones prestacions, es va escollir un dels microcontroladors més complets de la família, el PIC32MX795F512H *[4]*.

A la *Taula 9.1.* es detallen les seves principals característiques:

Especificació	**Valor**
Nombre de pins	64
Màxima Freqüència	80 MHz
Tensió d'alimentació	2,3 a 3,6 V
Memòria de programa (Flash)	512 KB
Memòria de dades (RAM)	128 KB
Temperatura de funcionament	-40 a 105 ºC
Mòdul de CAN	2
Mòdul de UART	5
Mòdul USB	1 (USB 2.0)

Taula 9.1. Especificacions del PIC32MX795F512H. Font: datasheet [4].

Com es pot observar, compleix sobradament totes les especificacions i millora considerablement les prestacions del PIC18. Cal destacar però, que no hi ha cap referencia a l'equip amb aquest microcontrolador. Això suposa un escenari nou amb canvis significatius. Per exemple, el software per fer la programació ha de ser el MPLAB X enlloc del MPLAB, utilitzat per l'equip i la universitat fins ara. Tot això implica haver de resoldre més problemes, però a l'hora porta molt valor afegit de cara a l'experiència de l'equip.

A la *Figura 9.4.* es mostra l'esquemàtic de l'àrea de control:

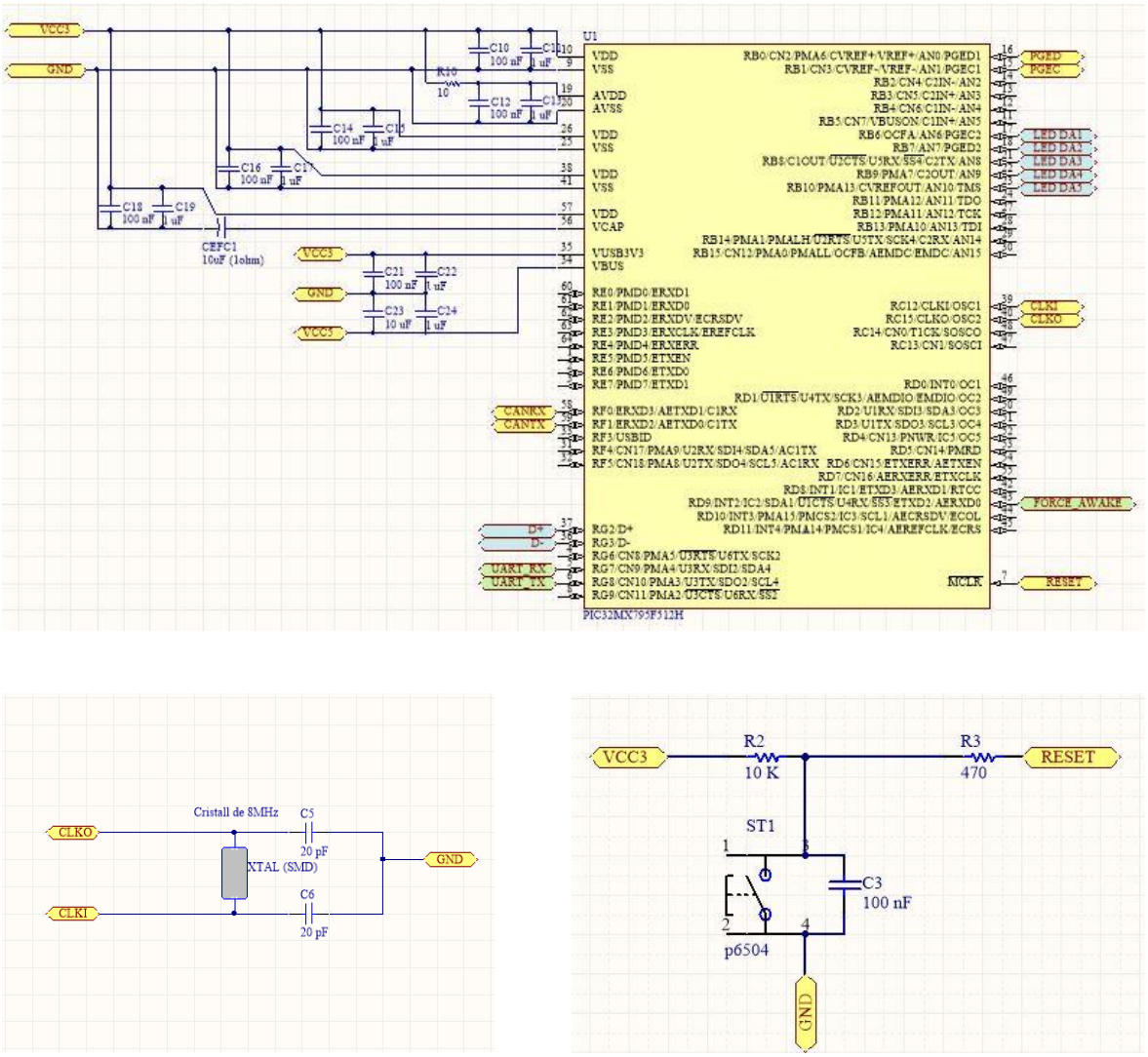


Figura 9.4. Esquemàtic de l'àrea de control. Font: pròpia.

Com es pot observar a l'esquemàtic, hi ha una sèrie de components que acompanyen al microcontrolador. Aquests s'han col·locat seguint les recomanacions del *datasheet*. Els components més importants a tenir en compte són:

- **Condensadors** (C10-C24, CEFC1 i R10): El seguit de condensadors que s'han col·locat tenen la funció d'estabilitzar la tensió d'alimentació del microcontrolador. És important que a l'hora de distribuir-los a la PCB, estiguin el més a prop possible del microcontrolador. Els valors dels diferents condensadors s'han extret del *datasheet [4]*.

- **Rellotge** (XTAL, C5 i C6): El microcontrolador necessita d'un senyal de rellotge extern per tal de poder funcionar a la freqüència màxima de 80 MHz. És per aquesta raó que s'ha instal·lat un cristall de quars de 8 MHz. El controlador ens permet multiplicar aquesta freqüència i és d'aquesta forma com aconseguim que el PIC treballi a 80 MHz. Seguint les indicacions del *datasheet [4]*. s'han col·locat un parell de condensadors, el valor del qual també ve determinat en funció de la freqüència del cristall.

- **Circuit RESET** (ST1, C3, R2): Aquest circuit permet *resetejar* el microcontrolador en cas de que sigui necessari per mal funcionament. El condensador té la funció de filtrar possibles rebots al fer servir el polsador, els quals provocarien molts senyals de *reset*. La resistència R2 es fa servir com a *pull-up*, de tal forma que si no s'està polsant el polsador, aquell pin del PIC rep un senyal digital de nivell alt.

Finalment, també s'etiqueten els pins del microcontrolador necessaris per fer la programació. Per programar el microcontrolador s'utilitzarà una eina de la mateixa empresa que requereix la connexió de 5 pins. Aquests són: alimentació (VCC3), terra (GND), reset (RESET) i dos pins a través dels quals carrega el programa al microcontrolador (PGED i PGEC).

9.3. Àrea de comunicacions CAN

9.3.1. Què és el bus CAN?

El bus CAN (*Controller Area Network*) és un protocol de comunicació sèrie asíncron entre dispositius. Va ser dissenyat inicialment per a la industria de procés i actualment és el bus de comunicacions més utilitzat en automoció.

És un bus diferencial, és a dir, que a l'hora d'interpretar els polsos es basa en la diferència de potencial entre dos cables (*CAN High* i *CAN Low*). Enviant un seguit de polsos digitals a través d'aquests dos cables, s'aconsegueix compartir informació entre dos o més dispositius o centraletes. La principal avantatge i raó de ser d'aquest sistema és la reducció de l'afectació de les EMIs (*Electromagnetic Interferences*) sobre la informació que es transmet. Una pertorbació d'aquest tipus afecta d'igual forma a cadascun dels dos cables, de tal forma que idealment la diferència entre els dos no es veu afectada. D'aquesta forma, s'aconsegueix un sistema robust i fiable, amb un cablejat simple i immune a les interferències.

A la *Figura 9.5.* es pot veure una representació esquemàtica d'una xarxa compartida per diverses centraletes. S'ha de tenir en compte que cadascun dels extrems de la xarxa s'hi ha de col·locar una resistència terminadora de 120 Ω.

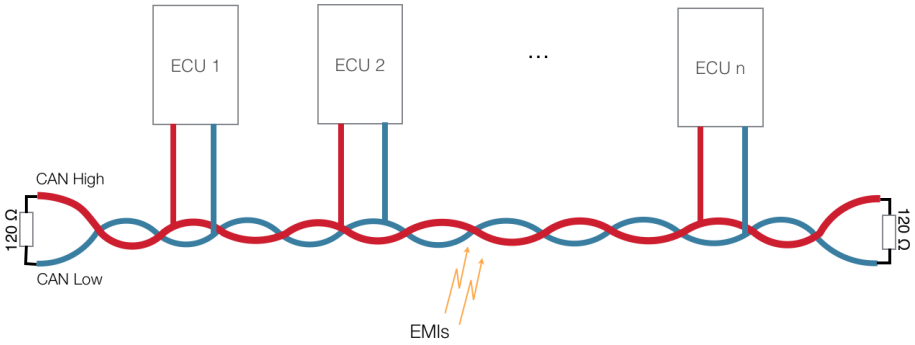


Figura 9.5. Representació d'una xarxa bus CAN. Font: [1].

Per tal de que hi hagi una bona transmissió i recepció de dades, s'han de configurar alguns paràmetres de la xarxa. Un dels paràmetres més importants és la velocitat de la xarxa, que s'expressa en bits per segon. Per identificar la informació i poder diferenciar entre els missatges, aquests s'estructuren d'una determinada forma. A grans trets, primer s'envia informació que identifica el missatge i seguidament s'envien les dades, que com a màxim estaran contingudes en un paquet de 8 bytes, que equivalen a 64 bits.

9.3.2. Distribució de la xarxa de bus CAN al CAT08e

La centraleta principal (dSPACE MicroAutobox II) del cotxe disposa de 4 xarxes de bus CAN. Les diferents centraletes i dispositius del vehicle es distribueixen en aquestes 4 xarxes amb la finalitat de no saturar-ne cap.

La *Figura 9.6.* mostra un esquema de la distribució i velocitat d'aquestes 4 xarxes amb els dispositius connectats a cadascuna.

Figura 9.6. Distribució de les xarxes bus CAN del CAT08e. Font: Pròpia.

9.3.3. Disseny de l'àrea de CAN

Pel que fa al disseny de la PCB, l'àrea de comunicacions CAN està basada fonamentalment en un component electrònic encarregat de fer d'intermediari entre la xarxa i el microcontrolador. Aquest component s'anomena **transceptor de CAN** (*Transceiver*) *[12]* i la següent figura mostra la configuració del circuit electrònic que l'envolta:

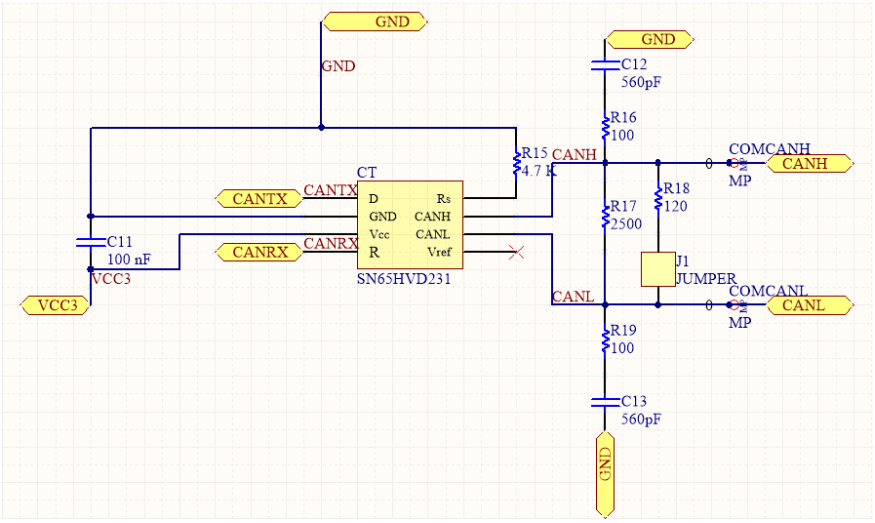


Figura 9.7. Esquemàtic de l'àrea de comunicacions CAN. Font: pròpia.

Aquest dispositiu s'alimenta a 3,3 V i també compta amb un condensador per estabilitzar-la. Com s'observa a l'esquemàtic, també s'hi col·loquen una sèrie de resistències segons les indicacions del *datasheet*. Finalment trobem un *jumper* (J1), que permet connectar la resistència de 120 Ω en cas que aquesta centraleta sigui un dels extrems de la xarxa.

9.4. Àrea de comunicació USB

Com ja s'ha explicat al punt *7.4 Sistema d'enregistrament*, la comunicació entre el microcontrolador i la unitat de memòria on s'enregistraran les dades serà mitjançant el protocol USB.

L'USB (*Universal Serial Bus*) és un estàndard universal que defineix el cablejat, els connectors i el protocol de comunicacions entre dispositius. Aquest protocol es basa en un bus de dades que es transmeten en sèrie i de forma asíncrona, és a dir, no caldrà senyal de rellotge. D'igual forma que el CAN, la comunicació és diferencial mitjançant dos únics cables, en aquest cas anomenats *D+* i *D-*.

A la *Figura 9.8.* es mostra l'esquemàtic de l'àrea de comunicacions USB:

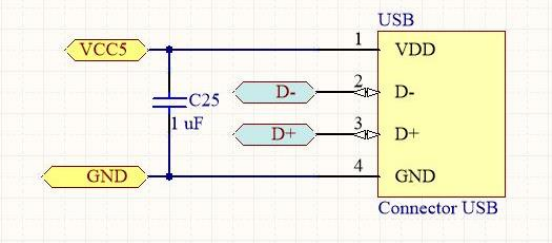


Figura 9.8. Esquemàtic de l'àrea de comunicacions USB. Font: pròpia.

Com s'observa, les connexions són molt simples. Bàsicament per establir la comunicació fan falta els dos pins (*D+* i *D-*) que es connecten a dos pins específics de la placa. També és necessari alimentar el dispositiu de memòria a 5 V. Per estabilitzar aquesta tensió d'alimentació es col·loca un condensador (C25), el valor del qual s'extreu del *datasheet [7]*.

Cal recordar que el connector USB també és estàndard, i això caldrà tenir-ho en compte a l'hora de situar i connectar el component a la placa. La *Figura 9.9.* mostra un connector USB del tipus A, que és el més comú i el que es farà servir en aquest projecte:

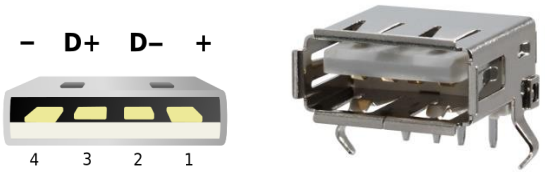


Figura 9.9. Connector USB tipus A. Font: google.com.

9.5. Àrea d'interfície externa

Per últim, la placa compta amb una sèrie d'elements que faciliten la interacció entre l'usuari i el mòdul. Aquests components són claus per analitzar i conèixer l'estat de funcionament en què es troba la centraleta. Principalment hi trobem dos tipus d'elements:

- **LEDs**: s'han col·locat 5 LEDs de diferents colors connectats directament a sortides digitals del microcontrolador. D'una forma molt simple i visual es pot conèixer l'estat en que es troba el programa i si tot està funcionant correctament.

- **Punts de mesura** (*measuring points*): són punts de la placa on és senzill mesurar-hi el voltatge o comprovar-ne el senyal. S'han establert diferents punts de mesura en els punts claus del circuit com són: les alimentacions a 12, 5 i 3 V, la seva referència a terra (*ground*), els dos pins de la xarxa CAN (*CAN High* i *CAN Low*) i els dos pins de la comunicació USB (*D+* i *D-*).

També es podrien incloure en aquesta àrea altres elements d'interacció comentats anteriorment, com són el **circuit *RESET*** (punt *9.2 Àrea de control*) i el **connector USB** (punt *9.4 Àrea de comunicació USB*) on l'usuari hi haurà de connectar la unitat de memòria.

A la *Figura 9.10.* es mostra l'esquemàtic de l'àrea d'interfície externa:

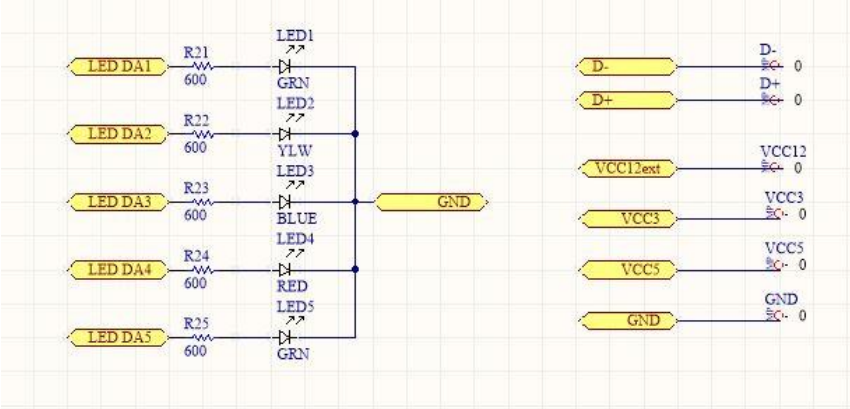


Figura 9.10. Esquemàtic de l'àrea d'interfície externa. Font: pròpia.

9.6. Disposició final de la placa

Un cop dissenyades totes les àrees, les següents figures mostren el resultat final de la PCB.

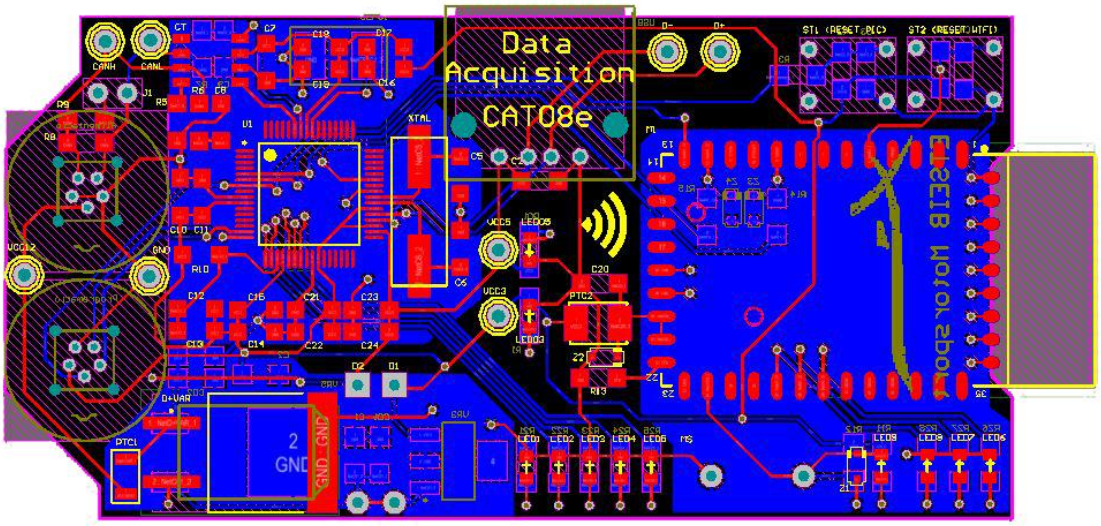


Figura 9.11. Representació de les capes i connexions de la PCB. Font: pròpia.

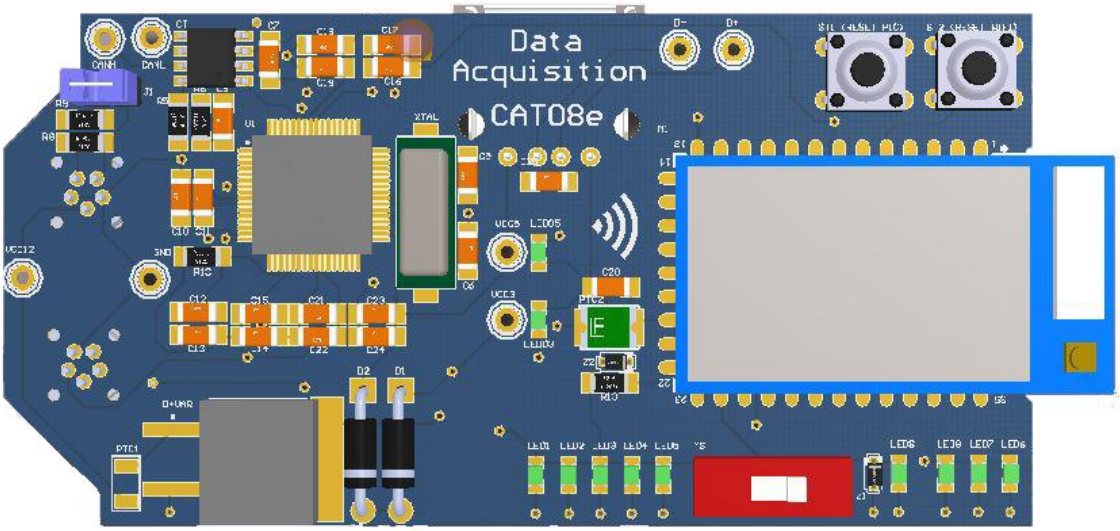


Figura 9.12. Representació en 3D de la part frontal de la PCB. Font: pròpia.

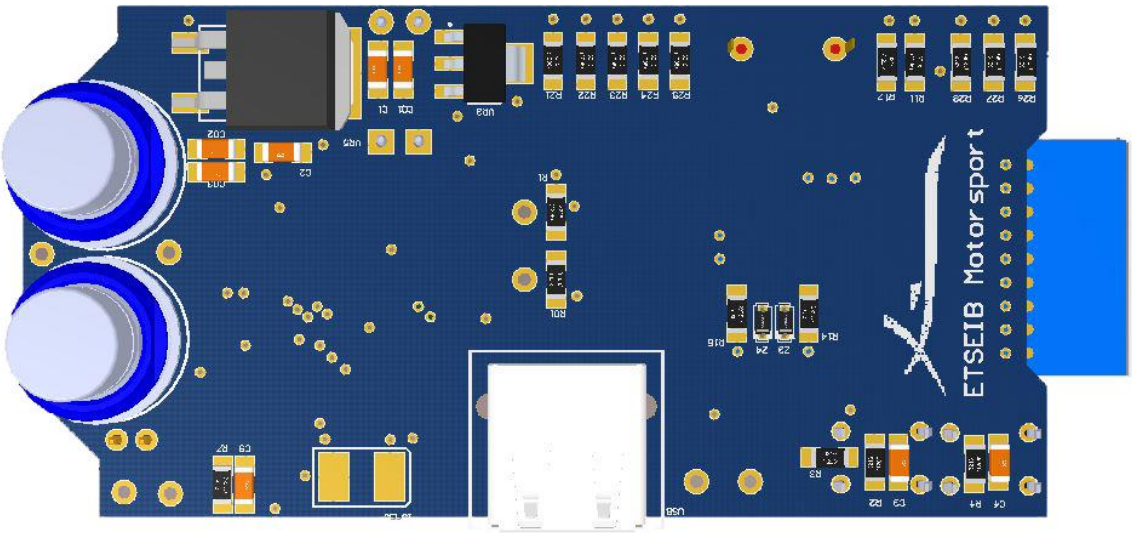


Figura 9.13. Representació en 3D de la part frontal de la PCB. Font: pròpia.

10. Desenvolupament del software

Per tal que el mòdul executi les funcions necessàries, és necessari programar el microcontrolador. Tot seguit es descriuran tots els procediments per fer-ho, així com l'estructura del programa principal.

10.1. Interfície de programació: *MPLAB X IDE*

El microcontrolador utilitzat és el PIC32MX795F512H de l'empresa Microchip. Aquesta empresa ofereix un entorn de desenvolupament propi per tal d'escriure, compilar i programar el codi. Abans de realitzar aquest projecte, tots els referents que es tenien dins de l'equip o la universitat, havien programat els microcontroladors utilitzant l'entorn MPLAB IDE. Amb el llançament dels PIC32, l'empresa també va incorporar una versió molt renovada de l'entorn, **MPLAB X IDE** *[6]*. Aquesta nova versió permet el desenvolupament del software de qualsevol microcontrolador de la companyia, però la versió antiga no és compatible amb els PIC32. Aquest fet obliga a fer servir aquesta nova versió del programa, i és aquí on trobem un dels reptes més importants del treball. La versió utilitzada per realitzar aquest projecte és la v2.20.

Algunes de les diferències més importants entre el MPLAB IDE i el MPLAB X IDE, són:

- Al MPLAB X, la **configuració** del dispositiu, el compilador i el programador, són propis de cadascun dels projectes. A la versió antiga, s'havien d'escollir globalment al programa i implicaven a tots els projectes oberts. Això facilita molt el treball simultani amb varis projectes ja que no cal tocar cap configuració al programar-los.
- La connexió amb l'**eina de programació**, amb la versió antiga, es pot fer inclús abans d'iniciar un projecte. En canvi, amb la nova versió, només estableix la connexió quan inicies el programador o el *debugger*, de tal forma que fins llavors no pots detectar possibles errors.
- Una de les avantatges més significatives del MPLAB X és que les **variables i funcions estan enllaçades** i pots accedir al fitxer on estan declarades polsant la tecla *Ctrl* al mateix temps que cliques sobre elles. A part d'això, en tot moment et dóna informació de si una funció o variable no la troba declarada. Aquesta característica, sens dubte, agilitza molt el procés de programació; sobretot en programes amb molts fitxers.
- El principal inconvenient de la nova versió és que obliga a fer servir un nou model de l'eina de programació, el cost de la qual és elevat. Això ha obligat a l'equip a adquirir el PICKit 3 *[5]* i a no poder utilitzar el PICKit 2, del qual ja es disposava. En quant al hardware, això no suposa cap canvi, ja que la connexió és idèntica.

Per tal de familiaritzar-se amb el nou entorn de programació i les seves particularitats, es va prendre la decisió d'adquirir el *PIC32 USB Starter Kit II [7]*. És un mòdul prefabricat que compta amb un PIC32 i permet executar-hi alguns programes ja dissenyats. Amb aquesta eina es van poder provar moltes de les funcions i aplicacions amb la comunicació USB. Existeix una extensa documentació i exemples d'aquests mòduls de proves per tal que l'usuari pugui tenir una referència de qualsevol aplicació.

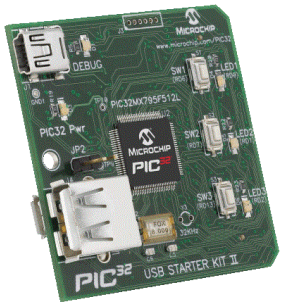


Figura 10.1. PIC32 USB Starter Kit II [7]. Font: microchip.com.

10.2. Paquet de llibreries: *MPLAB Harmony*

Un aspecte fonamental en el desenvolupament del software ha sigut el paquet de llibreries que ofereix l'empresa Microchip, anomenat **MPLAB Harmony** *[8]*. Aquesta plataforma facilita el desenvolupament del firmware d'una forma integrada, flexible i compatible amb qualsevol dispositiu. Inclou un conjunt de llibreries per a tots els tipus de perifèrics. S'ha de tenir en compte que aquesta plataforma només serveix per a dispositius de la família PIC32. Aquesta plataforma és totalment lliure i gratuïta i disposa d'una extensíssima documentació de cadascuna de les seves funcions i llibreries. L'empresa també facilita un espai de fòrum online que permet els desenvolupadors compartir experiències i resoldre dubtes.

De totes les llibreries del paquet, en aquest projecte seran de vital importància les següents:

- **CAN** (*plib_can*): inclou les funcions de transmissió i recepció de missatges de CAN. També conté funcions d'inicialització i configuració dels paràmetres de la xarxa.
- **USB** (*usb_host*): inclou les funcions per establir la connexió amb el dispositiu USB. En aquest projecte el microcontrolador actua com a *host* i la unitat de memòria com a dispositiu. És per això que utilitzem aquesta llibreria, tot i que també hi ha l'opció de fer funcionar el microcontrolador com a *device*.
- ***Timer*** (*sys_tmr*): inclou funcions per controlar els diversos *timers* del microcontrolador. Aquests funcionen com un comptador de temps a partir de la

senyal del rellotge. A partir dels *timers* es poden generar interrupcions al programa cada cert temps.

- ***File System*** (*sys_fs*): inclou totes les funcions d'interacció entre el microcontrolador i els fitxers, com són la creació, escriptura i tancament de fitxers.
- ***Device Configuration*** (*sys_devcon*): aquesta llibreria recull la informació del model de dispositiu assignat al projecte i deriva a la llibreria específica per aquell dispositiu. Amb aquestes llibreries s'estableix la connexió entre el software i el hardware del microcontrolador i altres funcions de baix nivell.
- ***Ports*** (ports_p32mx795f512h): inclou les funcions per configurar els ports del microcontrolador. Aquestes funcions són necessàries per activar i definir els ports com a sortides o entrades, digitals o analògiques. També defineix les funcions que permeten modificar el valor de la sortida del port.

Aquestes llibreries són essencials per poder desenvolupar el software del microcontrolador ja que permeten fer una programació d'alt nivell. A més de les funcions que ofereix la llibreria, també s'han creat algunes funcions per simplificar el programa principal i fer-lo més intel·ligible.

Cal destacar que en un d'aquests fitxers de les llibreries hi havia un error en el codi que va causar un important entrebanc en desenvolupament del programa. Finalment i gràcies a l'ajuda d'un fòrum de la web de l'empresa, es va localitzar i solucionar afegint una línia de codi a un d'aquests fitxers. Més endavant, Microchip va oferir una nova versió del paquet MPLAB Harmony, en la qual ja s'havia esmenat aquest error.

10.3. Estructura del programa principal

El programa principal es troba en el fitxer *main.c*. Aquest fitxer té un codi molt simple, on es criden només dues funcions. En primer lloc, s'executa una funció d'inicialització, on es configuren tots els paràmetres necessaris per al funcionament. En segon lloc, es crea un bucle i s'executen una sèrie de tasques de forma infinita. El codi és el següent:

```
SYS_Initialize();          %Inicialitzacions

while (true)               %Bucle infinit
{
    SYS_Tasks();           %Tasques del bucle infinit
}
```

A continuació es detallaran l'estructura i funcions més important d'aquestes dues comandes.

10.3.1. Inicialitzacions

10.3.1.1. Rellotge

Abans de fer la inicialització dels diferents mòduls o perifèrics del microcontrolador, cal definir els bits de configuració (*configuration bits*). Aquests paràmetres permeten configurar certs aspectes de funcionament adequant-los a les necessitats de l'aplicació. Els paràmetres més importants són aquells que tenen a veure amb el funcionament del rellotge.

Com s'ha comentat al punt *9.2 Àrea de control*, el controlador ha de funcionar amb un rellotge extern que emet una senyal de 8 MHz. Per tal de que pugui realitzar les tasques a la màxima freqüència, el microcontrolador permet multiplicar aquesta freqüència mitjançant el que es coneix com a PLL (*Phase-Locked Loop*). Com que es desitja que la freqüència de funcionament sigui de 80 MHz, caldrà multiplicar la senyal d'entrada per 10 (o el que és el mateix: multiplicar-la per 20 i dividir-la entre 2). La senyal de rellotge de referència, el seu mode de treball i els factors de multiplicació d'aquesta senyal, són uns dels aspectes a definir als bits de configuració; amb les següents línies de codi:

```
#pragma config FNOSC =     PRIPLL        %Rellotge Primari amb PLL
#pragma config FSOSCEM =   OFF           %Inhabilitació del rellotge Secundari
#pragma config POSCMOD =   XT            %Configuració del rellotge Primari
#pragma config FPLLIDIV =  DIV_2         %Divisor PLL (%2)
#pragma config FPLLMUL =   MUL_20        %Multiplicador PLL (x20)
```

10.3.1.2. Configuració dels ports

A continuació s'han de configurar els ports GPIO que han de complir una funció específica, com per exemple aquells que controlen els LEDs. Com s'ha explicat al punt *0*

Àrea d'interfície externa, s'han instal·lat 5 LEDs que tenen la funció de informar sobre l'estat del programa. Cadascun d'aquests està connectat a una pota del microcontrolador, per tant, aquests pins s'han de definir com a sortides (*outputs*) digitals per tal de poder controlar el seu valor. Posteriorment, en el moment d'inicialitzar el mòdul, també es fixen aquestes sortides a 0 per tal d'apagar tots els LEDs. Totes aquestes funcions d'inicialització s'agrupen dins la funció *BSP_Initialize()* i s'implementen amb les següents línies de codi:

```
/* Configurar els 5 ports com a sortides */
PLIB_PORTS_PinDirectionOutputSet(PORTS_ID_0, PORT_CHANNEL_B, BSP_LED1)
PLIB_PORTS_PinDirectionOutputSet(PORTS_ID_0, PORT_CHANNEL_B, BSP_LED2)
...
/* Posar a 0 totes aquestes sortides */
PLIB_PORTS_PinClear(PORTS_ID_0, PORT_CHANNEL_B, BSP_LED1)
...
```

Els ports connectats als LEDs són el 6, 7, 8, 9 i 10 del canal B. Per tal de fer més llegible el programa, s'han assignat els números de port a 5 variables, de tal forma que es pugui modificar de forma senzilla si es modifiqués el disseny hardware. D'igual forma, al mateix fitxer s'han definit les funcions *BSP_LEDOn* i *BSP_LEDOff* que contenen les funcions d'encendre (*PLIB_PORTS_PinSet*) i apagar (*PLIB_PORTS_PinClear*) els LEDs.

10.3.1.3. Inicialització del CAN

Un cop coneguda la freqüència de treball del microcontrolador (80 MHz) i la velocitat de la xarxa de CAN (1000 kbps), s'han de configurar una sèrie de paràmetres per tal que emissor i receptor s'entenguin. Per al càlcul d'aquests paràmetres, s'ha utilitzat un software gratuït anomenat *CAN Bit Timing Calculator*. Els valors obtinguts d'aquests paràmetres han estat els següents:

Propagation Time =	1 TQ
Phase Segment 1 (PS1) =	3 TQ
Phase Segment 2 (PS2) =	3 TQ
Syncro Jump Width (SJW) =	1 TQ
Baud Rate Prescale (BRP) =	5

El codi per implementar les funcions d'inicialització de la xarxa de CAN és el següent:

```
/* Habilitar el mòdul de CAN */
PLIB_CAN_Enable(CAN_ID_1);
...
/* Configurar paràmetres de la xarxa */
PLIB_CAN_PropagationTimeSegmentSet(CAN_ID_1, CAN_TIME_SEGMENT_LEN_1Q);
PLIB_CAN_PhaseSegment1LengthSet(CAN_ID_1, CAN_TIME_SEGMENT_LEN_3Q);
PLIB_CAN_PhaseSegment2LengthSet(CAN_ID_1, CAN_TIME_SEGMENT_LEN_3Q);
PLIB_CAN_SyncJumpWidthSet(CAN_ID_1, CAN_TIME_SEGMENT_LEN_1Q);
PLIB_CAN_BaudRatePrescaleSet(CAN_ID_1, CAN_BAUD_RATE_PRESCALE_5);
```

De la mateixa forma que abans, enlloc de posar-hi els valors directament, s'han definit unes variables al fitxer de configuració i així s'aconsegueix que sigui molt més àgil si en un futur es desitja modificar-les.

Tot seguit, també cal definir els canals per rebre o enviar missatges. Aquest microcontrolador permet tenir fins a 32 canals. Per cadascun dels canals s'ha de definir si serà de transmissió o de recepció i la mida del *buffer*, és a dir, el nombre màxim de missatges que emmagatzemarà a la cua (fins a un màxim de 32).

Un altre aspecte que es pot configurar són els filtres per hardware dels missatges. Consisteix en filtrar els missatges de tal forma que només entrin al *buffer* aquells que tinguin un identificador (o un rang) concret. Això és molt interessant i útil per no haver de processar tots els missatges que s'envien per la xarxa i fer-ho només amb aquells que interessen. Tal

com s'ha comentat al punt *7.3 Dades a recollir*, les dades s'envien en 7 missatges de CAN diferents. Com que es comparteix la xarxa amb altres centraletes, interessa que només entri un determinat rang de missatges, en aquest cas els que tinguin un identificador comprès entre el 512 i el 519 en notació decimal. Els identificadors estàndard de CAN tenen una longitud de 11 bits, tot i que es poden utilitzar identificadors en forma extensa amb longitud de 29 bits.

Així doncs, es configuraran els següents 2 canals:

- 1 canal de transmissió amb una mida del *buffer* de 8. Només s'haurà d'enviar un missatge per comunicar a la centraleta principal que el mòdul ja està llest per guardar dades.
- 1 canal de recepció amb una mida del *buffer* de 16. Se li aplica un filtre per l'identificador 512 (en binari, 0b01000000000) amb una màscara de valor 0b11111111000. D'aquesta forma, aconseguim que només entrin al *buffer* aquells missatges on els 8 primers bits es corresponguin amb els 8 primers bits del número 512. Així doncs, només entraran els missatges compresos entre el 0b01000000000 i el 0b01000000111 (en notació decimal, entre el 512 i el 519).

Per definir i configurar aquests canals, s'ha implementat amb el següent codi:

```
/* Configuració canal transmissió */
PLIB_CAN_ChannelForTransmitSet(CAN_ID_1, CAN_CHANNEL1, 8, ...);

/* Configuració canal recepció */
PLIB_CAN_ChannelForReceiveSet(CAN_ID_1, CAN_CHANNEL0, 16, CAN_RX_FULL_RECEIVE);
PLIB_CAN_FilterConfigure(CAN_ID_1, FILTER0, 0b01000000000, CAN_SID);
PLIB_CAN_FilterMaskConfigure(CAN_ID_1, FILTER_MASK0, 0b11111111000, ...);
PLIB_CAN_FilterToChannelLink(CAN_ID_1, FILTER0, FILTER_MASK0, CAN_CHANNEL0);
PLIB_CAN_FilterEnable(CAN_ID_1, FILTER0);
```

Tant les funcions d'inicialització i configuració de la xarxa com dels canals estan contingudes dins la funció *CAN_Initialize()*.

10.3.2. Bucle infinit

Les tasques contingudes dins del bucle infinit representen el cos principal del software. El programa s'estructura com una màquina d'estats. Cada estat representa una etapa del programa en la qual es realitzen una sèrie de funcions. En alguns casos, s'han de complir algunes condicions per tal de canviar a un altre estat. Al ser un bucle infinit, anomenarem aquests estats com "estats d'espera", ja que el programa romandrà allà fins que es

compleixin les condicions necessàries. En el cas que un programa executi certes funcions i canviï directament d'estat sense esperar cap condició, parlarem d' "estats de transició".

La *Figura 10.2*. mostra el diagrama d'estats del programa principal amb les corresponents condicions de transició:

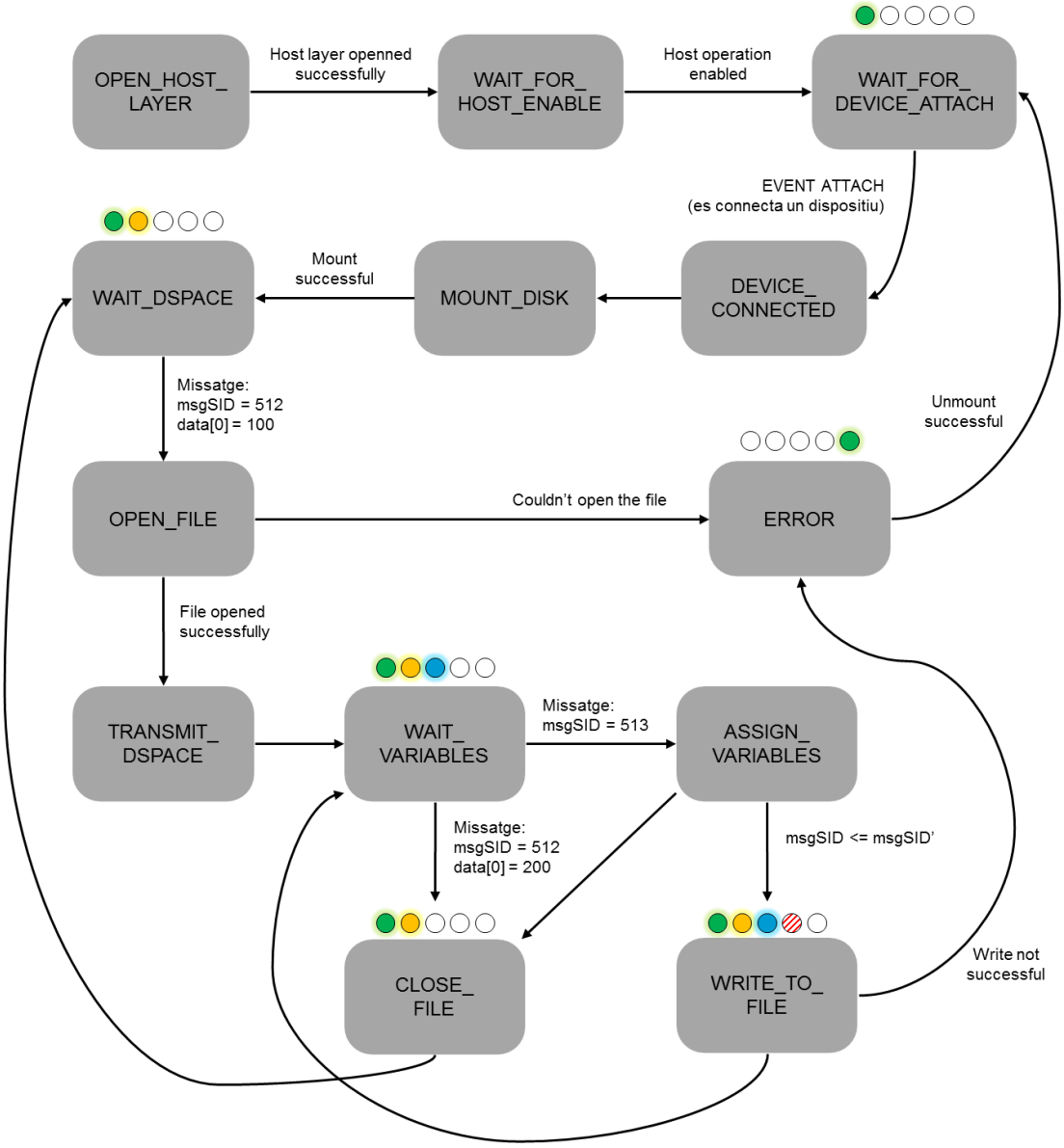


Figura 10.2. Diagrama d'estats del programa principal. Font: pròpia.

Dins dels diferents estats, es duen a terme les següents tasques:

- **OPEN_HOST_LAYER** i **WAIT_FOR_HOST_ENABLE**. En els dos primers estats s'executen funcions per definir les condicions necessàries per establir la comunicació USB i configurar el microcontrolador com a *host*.

- **WAIT_FOR_DEVICE_ATTACH**. En aquest estat no s'executa cap funció, simplement és un estat d'espera. A través d'una interrupció, es detecta quan s'ha connectat un dispositiu (*Event attach*) i és aleshores quan canvia al següent estat.
  El primer LED (verd) indica que s'ha assolit aquest estat.
- **DEVICE_CONNECTED**. És l'estat on va a parar el programa després de connectar-li un dispositiu USB. És un estat de transició que simplement deriva al següent estat.
- **MOUNT_DISK**. En aquest estat estableix la connexió amb la unitat de memòria per poder escriure o llegir des d'un directori.
- **WAIT_DSPACE**. És un estat d'espera. El programa espera a que s'enviï un cert missatge des de la centraleta principal, conforme aquesta està apunt per enviar dades. El missatge ha de tenir l'identificador 512 i el primer byte de dades ha de tenir valor igual a 100. La resta de bytes del missatge contenen informació de l'any, mes, dia, hora i minut, a partir dels quals es crearà el nom del fitxer. En aquestes condicions, es crearà una cadena de text que contindrà el nom del fitxer i s'anirà al següent estat.
  El segon LED (groc) indica que s'ha assolit aquest estat.
- **OPEN_FILE**. Aquest estat només té la tasca de crear un fitxer amb el nom corresponent i mantenir-lo obert. En cas que aquesta funció doni error, es deriva cap a l'estat ERROR. Si no hi ha cap problema, es canvia al següent estat amb normalitat.
- **TRANSMIT_DSPACE**. Un cop s'ha creat el fitxer, s'envia un missatge d'identificador 100 a la centraleta principal informant-la que el mòdul ja està preparat per començar a guardar. En aquest estat només s'executa aquesta tasca abans de passar al següent.
- **WAIT_VARIABLES**. En aquest estat, el programa espera fins que rep el primer missatge de dades (amb identificador 513). És important diferenciar aquest missatge de la resta ja que aquest conté la variable temps, comuna per totes les variables d'un mostreig. Quan rep aquest missatge, fa l'assignació de variables i canvia d'estat. En cas de rebre el missatge amb identificador 512 i primer byte igual a 200, es deriva a l'estat CLOSE_FILE amb l'objectiu de tancar el fitxer.
  El tercer LED (blau) indica que s'ha assolit aquest estat.
- **ASSIGN_VARIABLES**. Un cop rebut el primer missatge, es deriva a un estat on es processen la resta de missatges (d'identificadors entre 514 i 519) i s'assignen els valors de cadascuna de les variables. Com que els missatges s'envien per ordre, quan es rep un missatge amb identificador més petit que l'anterior, es passa al següent estat. En cas de rebre el missatge amb identificador 512 i primer byte igual a 200, es deriva a l'estat CLOSE_FILE amb l'objectiu de tancar el fitxer.
- **WRITE_TO_FILE**. Un cop rebudes totes les dades, es procedeix a escriure-les al fitxer de text. En cas que aquesta funció doni error, es deriva cap a l'estat ERROR.

Si no hi ha cap problema, es retorna a l'estat WAIT_VARIABLES per esperar el següent bloc de dades.

El tercer LED (vermell) indica de forma intermitent, encenent-se i apagant-se cada certs cops que s'ha complert el cicle de rebre i escriure un conjunt de dades.

- **CLOSE_FILE**. Aquest estat s'assoleix quan la centraleta envia un missatge concret i té la única tasca de tancar el fitxer per parar-hi d'escriure. Quan s'ha tancat, s'apaguen els LEDs 3 i 4 i es torna a l'estat WAIT_DSPACE.
- **ERROR**. Cada cop que una funció respon de forma insatisfactòria es va a parar en aquest estat. En aquest punt, es prossegueix a desfer la connexió amb la unitat de memòria i tornar a l'estat WAIT_FOR_DEVICE_ATTACH.

  Per deixar constància que s'ha produït un error, en aquest estat s'apaguen tots els LEDs i s'encén només el cinquè LED (verd).

A l'*Annex B* es troben alguns fitxers amb la part principal del codi del software.

10.3.3. Esquema del programa

A mode de resum, la *Figura 10.3.* mostra un esquema del funcionament del programa:

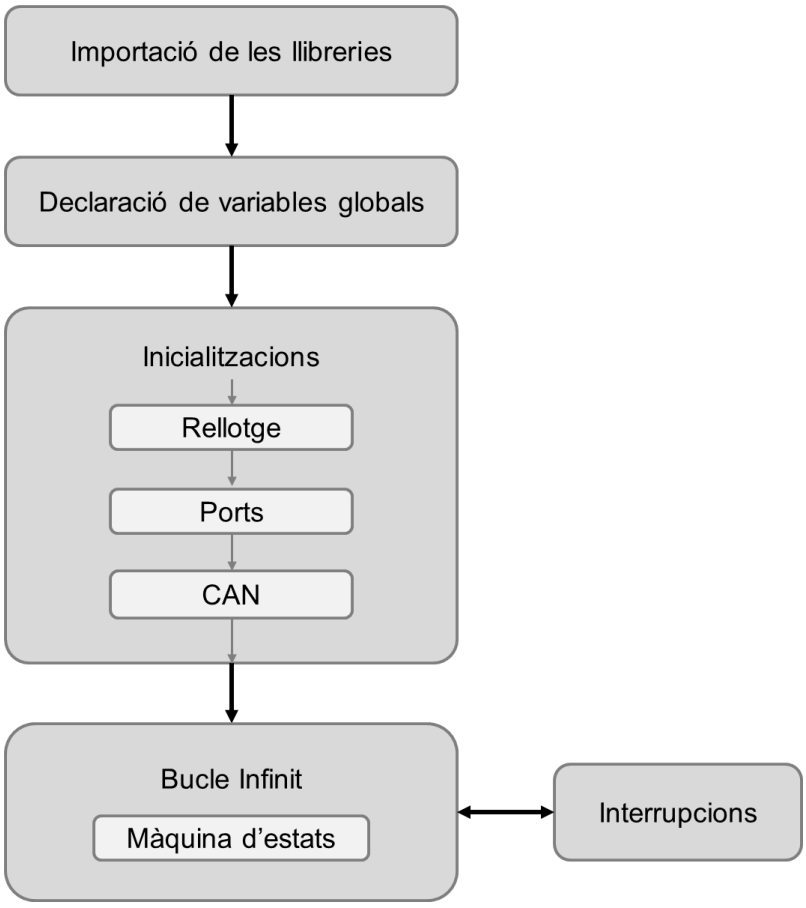


Figura 10.3. Esquema del programa principal. Font: pròpia.

10.4. Estructura de dades

Tal com s'ha comentat anteriorment, les 49 dades arriben al mòdul en 7 missatges de CAN amb identificadors compresos entre el 513 i el 519. La centraleta principal controla i actualitza totes les dades recollides en cadascuna de les altres centraletes. Amb una freqüència de **100 Hz** (cada **10 ms**), la centraleta principal fa una "fotografia" de totes les dades i les envia repartides en els diferents missatges. Per portar el control del temps, la primera variable que s'envia és un comptador que s'incrementa en una unitat cada cop que s'envia un bloc.

La distribució de les 49 dades en els missatges queda concretada a les següents taules:

Missatge 1 ID = 513	
Byte	Dada
0	TIME
1	FAIL
2	TMAX
3	CMAX
4	VMN5
5	CMN5
6	AUX1
7	AUX2

Missatge 2 ID = 514	
Byte	Dada
0	SUFR
1	
2	SUFL
3	
4	SURR
5	
6	SURL
7	

Missatge 3 ID = 515	
Byte	Dada
0	AC1X
1	AC1Y
2	AC1Z
3	VLON
4	
5	VTRA
6	DIST
7	

Missatge 4 ID = 516	
Byte	Dada
0	TQEN
1	TORQ
2	BRKF
3	BRKR
4	STEE
5	RPMM
6	RPMR
7	RPML

Missatge 5 ID = 517	
Byte	Dada
0	TMPM
1	TMPI
2	TMPT
3	ILVB
4	VLVB
5	I24V
6	I12V
7	SCLV

Missatge 6 ID = 518	
Byte	Dada
0	VBUS
1	
2	IBUS
3	VQ
4	IQ
5	VD
6	ID
7	SCHV

Missatge 7 ID = 519	
Byte	Dada
0	VMN1
1	VMN2
2	VMN3
3	VMN4
4	CMN1
5	CMN2
6	CMN3
7	CMN4

Taula 10.1. Contingut dels missatges de CAN. Font: pròpia.

Quan una dada requereix de més d'un byte per expressar el seu valor hi ha dos mètodes per separar el número:

- *Big-endian* (format Motorola): el byte de més pes es troba a l'inici.
- *Little-endian* (format Intel): el byte de més pes es troba al final.

Des de la centraleta principal totes les dades s'envien en ***little-endian***. Això s'ha de tenir en compte al programa a l'hora de reconstruir el valor de la variable.

Un cop totes les dades arriben al microcontrolador i aquest fa l'assignació de variables, només queda concatenar-les en una cadena de text i escriure-les al fitxer. Interessa que sigui d'una forma senzilla i pràctica per tal que després es pugui fer la lectura sense problemes. Per tal de facilitar les coses a la interfície (punt *11 Desenvolupament de la interfície gràfica*) es decideix separar cada variable amb un punt i coma "**;**" i fer un salt de línia després de cada bloc de dades.

Per tal de saber el nombre total de caràcters que suposa això, es diferencia entre els següents elements:

-	42 x Dades d'1 byte: valor entre 0 i 255 (3 dígits) ➔ 42 x 3	=	126 caràcters
-	7 x Dades de 2 bytes: valor entre 0 i 65535 (5 dígits) ➔ 7 x 5	=	35 caràcters
-	49 x Separadors de dades: ";" (1 caràcter) ➔ 49 x 1	=	49 caràcters
-	1 x Salt de línia: "/n/r" (2 caràcters) ➔ 1 x 2	=	2 caràcters
			212 caràcters

Per tant, cada 10 ms s'executarà la funció d'escriure una cadena de 212 caràcters al fitxer.

11. Desenvolupament de la interfície gràfica

L'últim pas per completar el projecte és el desenvolupament d'una interfície gràfica que extregui i tracti les dades del fitxer de text per tal de poder-les interpretar i analitzar.

La plataforma en la que s'implementarà aquesta interfície serà en Matlab-Simulink, ja que aquest ofereix una eina per a desenvolupar interfícies gràfiques: *GUIDE*. Amb aquesta eina també es va crear la interfície de l'any anterior.

La interfície gràfica es divideix, bàsicament, en dues parts:

- **Interfície de tractament de dades**. Aquesta interfície permet carregar-li un fitxer de text (amb l'estructura explicada al punt *10.4 Estructura de dades*) per extreure'n un fitxer de Matlab amb les dades tractades.
- **Interfície de visualització de dades**. Aquesta interfície permet visualitzar les dades a partir d'un fitxer de Matlab.

La primera part de la interfície ha sigut desenvolupada de nou, ja que requeria un tractament complet de les dades. La segona part, en canvi, ja va ser desenvolupada l'any anterior i no s'explicarà en aquest projecte. Si es desitja més informació d'aquesta part, es pot consultar el treball *[3]*.

11.1. Interfície de tractament de dades

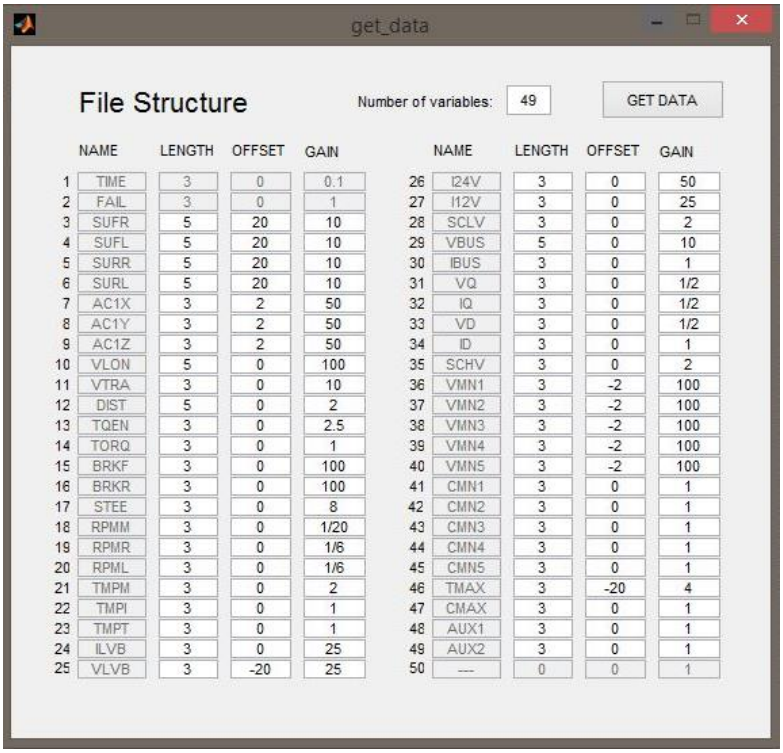
Com ja s'ha comentat al punt *7.3 Dades a recollir*, les dades han sigut tractades a la centraleta principal per tal de minimitzar el nombre de bytes (i per tant, missatges) necessaris per enviar-les totes. La principal finalitat d'això és alliberar de responsabilitats al mòdul. És per això, que la correcció (guany i desfasament) que se'ls hi aplica abans d'enviar-les, s'haurà de "desfer" a la interfície de l'ordinador.

Un altre objectiu era que el guany i desfasament aplicat a cadascuna de les dades fos editable. D'aquesta forma, si en algun moment es modifica a la centraleta principal, no és necessari re-programar la interfície, sinó que el propi usuari pot modificar-ho.

L'altre funció vital de la interfície és fer el càlcul absolut del temps. Cal recordar que la variable enregistrada (TIME) és un comptador de 0 a 200. S'ha dissenyat una funció que a partir d'aquests valors, els transforma a milisegons reals.

També es tracta la variable que conté informació de les dades binàries, de tal forma que en resulten 8 variables diferenciades amb valors de 0 o 1 per a cada instant.

L'aspecte de la interfície queda representat a la següent figura:



	NAME	LENGTH	OFFSET	GAIN		NAME	LENGTH	OFFSET	GAIN
1	TIME	3	0	0.1	26	I24V	3	0	50
2	FAIL	3	0	1	27	I12V	3	0	25
3	SUFR	5	20	10	28	SCLV	3	0	2
4	SUFL	5	20	10	29	VBUS	5	0	10
5	SURR	5	20	10	30	IBUS	3	0	1
6	SURL	5	20	10	31	VQ	3	0	1/2
7	AC1X	3	2	50	32	IQ	3	0	1/2
8	AC1Y	3	2	50	33	VD	3	0	1/2
9	AC1Z	3	2	50	34	ID	3	0	1
10	VLON	5	0	100	35	SCHV	3	0	2
11	VTRA	3	0	10	36	VMN1	3	-2	100
12	DIST	5	0	2	37	VMN2	3	-2	100
13	TQEN	3	0	2.5	38	VMN3	3	-2	100
14	TORQ	3	0	1	39	VMN4	3	-2	100
15	BRKF	3	0	100	40	VMN5	3	-2	100
16	BRKR	3	0	100	41	CMN1	3	0	1
17	STEE	3	0	8	42	CMN2	3	0	1
18	RPMM	3	0	1/20	43	CMN3	3	0	1
19	RPMR	3	0	1/6	44	CMN4	3	0	1
20	RPML	3	0	1/6	45	CMN5	3	0	1
21	TMPM	3	0	2	46	TMAX	3	-20	4
22	TMPI	3	0	1	47	CMAX	3	0	1
23	TMPT	3	0	1	48	AUX1	3	0	1
24	ILVB	3	0	25	49	AUX2	3	0	1
25	VLVB	3	-20	25	50	---	0	0	1

Figura 11.1. Pantalla de la interfície de tractament de dades. Font: pròpia.

Com s'observa, permet editar el desfasament (*offset*) i el guany (*gain*) de cadascuna de les dades. Al prémer el botó "GET DATA" obre una pantalla per seleccionar el fitxer de text i comença a fer el processament. Acabat el processament, obre un altre panell per guardar el fitxer de Matlab.

Aquest fitxer conté la informació de cadascuna de les variables emmagatzemades en un vector, és a dir, una llista de valors per a cada instant de temps. Una altra funcionalitat útil que s'ha incorporat és que, a l'hora de guardar el fitxer, el nom proposat està compost per la data (dia i hora) i la duració total de les dades enregistrades. Així és molt més fàcil identificar a quina cursa o prova correspon el fitxer.

12. Muntatge i validació

A l'hora de fer el muntatge i la programació del mòdul s'han dissenyat una sèrie de proves o estratègies per anar validant i comprovant el seu correcte funcionament. Aquestes proves es poden dividir en dos grans blocs: proves de hardware i proves de software.

12.1. Proves de hardware

Les proves de hardware s'inicien en el moment en que el fabricant envia la placa i durant el procés de soldadura. Aquestes proves són d'extrema importància per tal d'assegurar el correcte funcionament de cadascuna de les àrees de la placa i evitar possibles problemes futurs.

El primer que cal soldar i comprovar és l'**àrea d'alimentació**, ja que d'aquesta depenen la majoria de components. Així doncs, només es solden a la placa els components d'aquesta zona i es realitzen les següents proves:

- **Continuïtat**. Es comprova la continuïtat entre les pistes i components per validar que els circuits han estat ben dissenyats i la placa ben fabricada.
- **Alimentació**. S'alimentarà la placa a una tensió de 12 V que és a la que haurà de funcionar. En aquesta situació, es comproven les tensions de 5 V i 3,3 V a les sortides dels reguladors i a tots aquells punts per on s'han d'alimentar altres components. També s'ha de validar que tots els LEDs indicadors s'encenguin. Cal recordar que els reguladors admeten un rang de tensió d'entrada restringit entre 7,5 i 25 V (tenint en compte les limitacions de tots dos). Es comprova, també, que per tensions d'alimentació dins d'aquest rang, els voltatges a la sortida es mantenen als valors esperats.
- **Sobretensió d'alimentació**. El sistema està dissenyat per tal que arribin 12 V a la placa, però les bateries que alimenten l'electrònica poden arribar a proporcionar una tensió de 28 V. Reproduint el pitjor dels casos, s'ha alimentat durant uns instants a aquesta tensió i els components han resistit, tot i escalfar-se considerablement.
- **Sobrecorrent d'alimentació**. Alimentant la placa a 12 V s'ha provocat un curtcircuit entre les sortides dels reguladors i el terra (*ground*) provocant un sobrecorrent. Gràcies a l'escalfament que ha experimentat el fusible resetejable (o PTC) el corrent ha caigut en picat. Quan, en condicions normals d'alimentació, ha tornat a la temperatura de funcionament, tot ha tornat a la normalitat.
- **Pics de tensió**. La darrera prova amb l'àrea d'alimentació ha consistit en generar petits pics de tensió a l'alimentació i veure com aquests afectaven a l'entrada del

regulador. S'ha comprovat que gràcies a l'encapsulat de díodes, aquests pics de tensió han quedat amortits a l'entrada dels reguladors.

Amb l'alimentació soldada i testejada, s'ha continuat amb la soldadura de la resta de components. Amb tots els components soldats, s'han fet una sèrie de proves per comprovar que tot fos correcte:

- **Programació del microcontrolador**. Per tal d'assegurar que les potes de programació estaven ben connectades i que el funcionament del microcontrolador era correcte, s'ha programat un programa molt senzill per comprovar que no hi havia problemes durant aquest procés.
- **Cristall**. Amb l'ajut d'un oscil·loscopi, també s'ha comprovat que el cristall de quars emetia un senyal amb una freqüència determinada.
- **CAN**. Per validar el funcionament del transceptor de CAN s'ha hagut d'elaborar un programa que simplement enviés un missatge de forma continua. Utilitzant un lector de xarxes CAN i l'oscil·loscopi, s'ha verificat que el missatge s'estava transmeten de forma correcta.

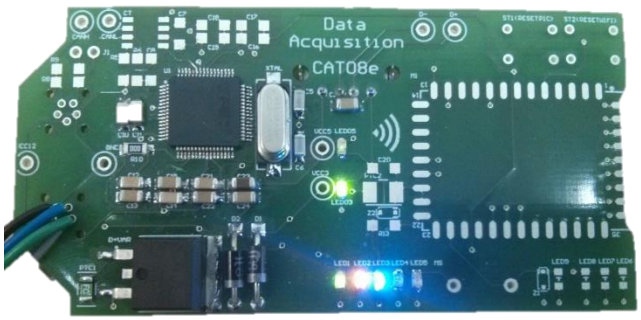


Figura 12.1. Muntatge de les àrees d'alimentació i control. Font: pròpia.

12.2. Proves de software

Per encarar la verificació del software s'ha fet seguint un procés de divisió de funcions. S'han anat dissenyant i elaborant funcions específiques i, a mesura que s'anaven validant, s'afegien funcionalitats més complexes. A grans trets, les funcions que s'han provat han sigut les següents:

- **Encendre LEDs**. El primer que es comprova és que el microcontrolador sigui capaç de controlar les sortides on té connectats els LEDs. D'aquesta forma no tant sols es comprova el funcionament d'aquestes sortides, sinó que també es valida el bon funcionament del microcontrolador i el seu rellotge.

- **Enviar missatge CAN**. Tal com s'ha comentat al punt anterior, s'ha elaborat un programa que simplement envia missatges de CAN. D'aquesta forma, s'aconsegueix validar la configuració de la xarxa.
- **Crear i escriure al fitxer**. També s'ha dissenyat un programa que només tenia la tasca de crear un fitxer i escriure-hi una cadena de text curta ("Hello World!"). Amb aquest procés es validava tota la comunicació necessària entre el microcontrolador i la unitat de memòria.

Amb aquestes funcions desenvolupades i testejades, es va desenvolupar la primera versió del programa final, la qual ha anat experimentant certes millores i evolucions fins a arribar al software definitiu exposat al punt *10.3 Estructura del programa* principal.

Per realitzar la versió final del software i totes les proves ha sigut imprescindible l'ús del depurador (*debugger*). El PICKit 3 permet executar el programa en "mode depurador" i d'aquesta forma es pot controlar el punt en què es troba el programa i el valor de totes les variables i registres en aquell instant. D'aquesta forma és molt més senzill detectar els errors per poder-los corregir.

Un cop s'havia programat el microcontrolador amb el software definitiu també era necessari verificar que les dades enregistrades i la freqüència de guardat eren les correctes. Mitjançant el lector de xarxes CAN de l'empresa Kvaser es van poder realitzar les proves pertinents. Aquesta eina permet generar un trànsit de missatges configurant molts aspectes. Bàsicament es fan proves amb un flux de les següents característiques:

- Període d'enviament d'1 ms (nombre total de missatges: 10000).
- Identificadors entre el 513 i el 519 per ordre (escombrat).
- Bytes de dades a mode de comptador.

Figura 12.2. Kvaser Leaf Light HS v2 [2]. Font: kvaser.com.

D'aquesta forma es podia verificar que les dades guardades eren les correctes.

Finalment, amb el software testejat, es passa a la fase de comunicar-lo amb la resta de centraletes del cotxe. En primer lloc, es prova el mòdul només connectat a la centraleta principal. Progressivament s'hi van afegint la resta de centraletes i es comença a fer variar els valors dels diferents sensors, per comprovar que el sistema enregistra aquests canvis. En última instància, quan ja funciona amb tota l'electrònica del cotxe, es fa la darrera prova. Aquesta consisteix en deixar el sistema en funcionament durant més de 40 minuts, que és la durada de la prova més llarga de la competició. Al cap de gairebé una hora, el mòdul segueix funcionant amb normalitat.

Tot i així, cal anar verificant el seu funcionament constantment. Quan el cotxe estigui a punt per anar a provar-lo en cursa, caldrà veure com el mòdul respon a la situació i comprovar que les dades enregistrades són lògiques i correctes.

12.3. Assemblatge final

Un requeriment de la centraleta és que sigui estanca. És per això que s'ha hagut de dissenyar una forma que la placa quedés protegida. La solució final ha consistit en adquirir una caixa comercial d'alumini de dimensions 100x50x21 mm.

Figura 12.3. Caixa Hammond 1590GBK. Font: hammondmfg.com

Per poder fer les connexions, s'han situat dos connectors de l'empresa LEMO. Un connector conté les connexions necessàries per programar el microcontrolador, i l'altre les connexions necessàries pel funcionament (alimentació i connexió a la xarxa CAN).

A part dels dos forats pels connectors, també s'ha hagut de fer un forat per poder-hi connectar la unitat de memòria i un altre per treure l'antena de telemetria.

En primera instància es va fer el disseny CAD de tota la centraleta. Les següents figures mostren el resultat d'aquest disseny:

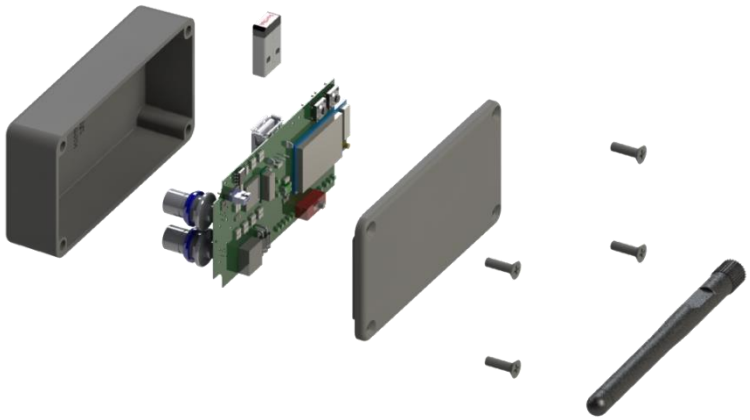


Figura 12.4. Vista explosionada de la centraleta. Font: pròpia.

Figura 12.5. Disseny CAD de la centraleta. Font: pròpia.

El resultat final, un cop fets els forats i amb tot l'assemblatge fet, ha estat el següent:

Figura 12.6. Centraleta d'enregistrament de dades i telemetria. Font: pròpia.

13. Pressupost econòmic

En aquest apartat es fa un estudi dels costos de la realització del projecte. Gran part dels recursos i material per dur-lo a terme han estat finançats pels múltiples patrocinadors de l'equip. Tot i així, amb l'objectiu de fer un anàlisi i pressupost real, s'ha comptabilitzat tot.

Es desglossaran els costos en 4 blocs: costos de material, d'equipament, de software i de personal.

13.1. Costos de material

Dins dels costos de material s'inclou el preu dels components necessaris per a fabricar la placa, és a dir, el material fungible. A partir del BOM (*Bill of* Materials) es pot calcular de forma senzilla els costos totals dels components. En aquest pressupost no s'han tingut en compte aquells components de l'àrea de telemetria.

A la taula següent es detallen els costos de tot el material:

Component	**Quantitat** [unitats]	**Cost unitari** [€/unitat]	**Cost** [€]
Condensador (1206) – 10 uF	4	0,138	0,552
Condensador (1206) – 1 uF	10	0,091	0,910
Condensador (1206) – 100 nF	9	0,097	0,873
Condensador (1206) – 20 pF	2	0,059	0,118
Condensador (1206) – 560 pF	2	0,054	0,108
Condensador resistiu (CEFC) – 10 uF, 1Ω	1	7,48	7,48
Resistència (1206) – 10 Ω	1	0,091	0,091
Resistència (1206) – 100 Ω	1	0,091	0,091
Resistència (1206) – 120 Ω	1	0,091	0,091
Resistència (1206) – 220 Ω	1	0,091	0,091
Resistència (1206) – 470 Ω	2	0,091	0,182
Resistència (1206) – 600 Ω	5	0,091	0,455
Resistència (1206) – 2,5 Ω	1	0,091	0,091
Resistència (1206) – 4,7 kΩ	1	0,091	0,091
Resistència (1206) – 10 kΩ	1	0,091	0,091
LED Verd (0805)	4	0,097	0,388
LED Groc (0805)	1	0,097	0,097
LED Vermell (0805)	1	0,097	0,097
LED Blau (0805)	1	0,097	0,097
Cristall de quars – 8 MHz	1	0,236	0,236

Jumper	1	0,115	0,115
Fusible PTC (MC33191)	1	0,168	0,168
Polsador (Switch Tact)	1	0,220	0,220
Transceptor de CAN (SN65HVD231)	1	2,08	2,08
Díode + Varistor (RBO08)	1	2,35	2,35
Díode (1N4004)	2	0,136	0,272
Regulador de tensió 3,3 V	1	0,543	0,543
Regulador de tensió 5 V	1	0,561	0,561
Connector LEMO 305	1	105,11	105,11
Connector LEMO 307	1	114,85	114,85
Connector USB	1	0,706	0,706
Microcontrolador (PIC32MX795F512H)	1	7,94	7,94
Memòria USB 4GB	1	6,99	6,99
Caixa HAMMOND	1	13,72	13,72
Placa de circuit imprès (PCB)	1	290	290
TOTAL			**557,86 €**

Taula 13.1. Costos de material. Font: pròpia.

El cost total del material és de **557,86 €**.

13.2. Costos d'equipament

En aquest apartat s'inclouen tots els costos derivats dels equipaments que s'han hagut d'utilitzar pel desenvolupament del projecte. En cadascun d'ells s'hi ha aplicat l'amortització corresponent al temps que ha durat el projecte, que ha sigut d'1 any.

A la taula següent es detallen els costos d'equipament:

Concepte	Cost [€]	Amortització	Preu amortitzat
Ordinador personal	1.000	30%	300,00 €
Font d'alimentació (Rohde&Schwarz)	948	15%	142,20 €
Oscil·loscopi (Rohde&Schwarz)	798	15%	119,70 €
Kvaser Leaf Light V2	326,28	15%	48,94 €
Multímetre (PCE Instruments)	50	15%	7,5 €
PICKit 3	42,50	15%	6,38 €
USB Starter Kit II	44,95	30%	13,49 €
TOTAL			**638,21 €**

Taula 13.2. Costos d'equipament. Font: pròpia.

El cost total de l'equipament és de **638,21 €.**

13.3. Costos de software

Els costos de software fan referència als costos de les llicències del programari emprat per a realitzar el projecte. Les llicències són anuals i per tant no caldrà aplicar-hi amortitzacions.

A la taula següent es detallen els costos dels diferents softwares emprats:

Concepte	Descripció	Cost
Altium Designer 14	Disseny de circuits i layout de la placa.	6.795 €
MATLAB	Disseny de la interfície gràfica.	2.000 €
MPLAB X	Programació de microcontroladors del fabricant Microchip.	0 €
Compilador XC32	Compilador del codi per a microcontroladors de la família PIC32 de Microchip.	0 €
Paquet MPLAB Harmony	Paquet de llibreries per a microcontroladors de la família PIC32 de Microchip.	0 €
Kvaser CAN King	Lector i generador de missatges de CAN.	0 €
TOTAL		**8.795 €**

Taula 13.3. Costos de personal. Font: pròpia.

El cost total del software és de **8.795,00 €.**

13.4. Costos de personal

Finalment, en aquest apartat es valoraran econòmicament els costos de mà d'obra, estimant un sou per a cadascuna de les funcions a desenvolupar.

Les tasques a desenvolupar es poden agrupar en les següents:

- **Disseny del hardware**: cerca de components, disseny dels circuits electrònics i de la PCB. Se li assigna un sou de 50€/h.
- **Muntatge de la placa**: soldadura de tots els components i comprovació de les connexions. Se li assigna un sou de 30€/h.
- **Disseny de software**: desenvolupament i programació del codi del microcontrolador i de la interfície. Se li assigna un sou de 50€/h.
- **Control de qualitat**: disseny i execució del procés de validació i tests del mòdul. Se li assigna un sou de 50€/h.

A la taula següent es detallen els costos de personal:

Concepte	Cost horari [€/h]	Temps [h]	Cost total [€]
Disseny de hardware	50	200	10.000 €
Muntatge de la placa	30	30	900 €
Disseny de software	50	180	9.000 €
Control de qualitat	50	20	1.000 €
TOTAL		**430**	**20.900 €**

Taula 13.4. Costos de personal. Font: pròpia.

El cost total de personal és de **20.900,00 €**

13.5. Pressupost final

La *Taula 13.5.* mostra un resum de tots els costos del projecte i el cost total d'aquest:

Concepte	Cost
Costos de material	**557,86 €**
Costos d'equipament	**638,21 €**
Costos de software	**8.795,00 €**
Costos de personal	**20.900,00 €**
TOTAL	**30.891,07 €**

Taula 13.5. Cost total del projecte. Font: pròpia.

Així doncs, el pressupost aproximat del projecte és de **30.891,07 €**

Com es pot observar, bona part dels costos són costos de personal i de software. Això constata el fet que hi ha dos aspectes essencials per tal que el projecte es pugui desenvolupar. Per un costat, el suport dels patrocinadors que, a part de material, també proporcionen llicències de forma gratuïta. I per l'altre costat, la dedicació dels membres del projecte, que dediquen nombroses hores a cost zero. Sense aquests dos factors, el projecte no seria viable econòmicament, tenint en compte que només se n'ha de fabricar un prototip. En cas de fabricar-ne sèries més grans, els costos es veurien reduïts molt significativament.

14. Impacte mediambiental

L'anàlisi de l'impacte ambiental d'aquest projecte es focalitza sobretot en la fabricació del mòdul i dels materials i processos emprats per construir-lo.

La directiva europea RoHS (*Resitrictions of Hazardous Substances*) regula els materials i substàncies que es poden utilitzar en la fabricació de components elèctrics i electrònics. D'acord amb aquesta normativa es restringeix i es limita l'ús de certes substàncies (plom, mercuri, crom, cadmi i brom) que representen un perill per al medi ambient i requereixen processos de reciclatge complexes. Arrel d'aquesta directiva, els fabricants garanteixen que els seus components compleixen amb aquests requeriments.

Així doncs, tots els components muntats al mòdul i la PCB compleixen la directiva RoHS i així ho constata el segell de "RoHS compliant" a les seves fitxes tècniques.

Conclusions

L'objectiu inicial d'aquest projecte era fabricar un mòdul capaç d'enregistrar totes les dades del cotxe a una freqüència adequada per a poder fer-ne el posterior anàlisis. Es pot afirmar que l'objectiu ha estat assolit molt satisfactòriament. Les conclusions principals que se n'extreuen dels objectius assolits són:

- S'ha aconseguit una solució que **satisfà tots els requeriments inicials**. El mòdul és capaç d'enregistrar les dades a la freqüència requerida d'una forma fiable.
- El mòdul **millora les prestacions** de les solucions adoptades per l'equip en temporades anteriors.
- S'ha implementat satisfactòriament el propòsit d'**alliberar del màxim de responsabilitats** al mòdul. S'han transferit algunes de les responsabilitats a la centraleta principal i a la interfície de l'ordinador.
- S'ha fet un anàlisi profund de les dades a recollir i la resolució necessària. Mitjançant el **tractament de dades** s'ha aconseguit minimitzar el nombre de missatges emprats per enviar totes les variables.
- S'ha superat amb èxit el repte que suposava haver d'aprendre a fer servir **nou programari** per al desenvolupament del projecte i es valora molt positivament el coneixement adquirit.
- Aquest projecte dota a l'equip de coneixement sobre **noves eines i productes** mai utilitzats abans. S'ha fet ús d'un microcontrolador de la família PIC32 i de l'última versió del programa per desenvolupar el software, el MPLAB X IDE.
- També s'ha fet ús del paquet de llibreries MPLAB Harmony que ha permès estructurar el programa aconseguint un **codi comprensible** i amb moltes opcions de de cara al futur. El suport d'aquesta plataforma pot esdevenir un element clau per utilitzar aquests microcontroladors en la resta de centraletes del cotxe.

Per altre costat, hi ha alguns objectius que han quedat pendents degut a que no s'ha complert amb la planificació global del projecte i per tant no hi ha hagut temps d'executar-los. Aquests objectius que queden pendents, són:

- El mòdul encara no s'ha pogut provar en cursa ja que s'ha endarrerit la fabricació del cotxe. Per tant queda pendent **validar el mòdul en pista**. Tot i així, s'han pres totes les mesures per tal d'assegurar que el funcionament dins el cotxe sigui correcte.
- Amb el mòdul muntat i amb l'objectiu de garantir la fiabilitat de les dades, caldrà fer un bon **calibratge dels sensors** del cotxe.
- Un cop es facin les primeres proves amb el cotxe i durant tota la competició, caldrà analitzar i els fitxers guardats per verificar la **coherència de les dades** recollides.

Finalment es proposen vies de millora de cara a futurs dissenys:

- Tal com s'ha comentat al punt *9.1 Àrea d'alimentació*, substituir els reguladors de tensió per **convertidors CC/CC**. D'aquesta forma es millora l'eficiència i per tant es redueix el consum de la centraleta.
- Una bona via de millora per tal de conèixer el comportament del microcontrolador és dissenyar una sèrie de **proves d'estrès**. L'objectiu d'aquestes proves hauria de ser intentar trobar el temps que necessita el processador per a realitzar totes les funcions que se li requereixen i intentar trobar el seu límit. Per exemple, augmentar progressivament la freqüència d'enviament de missatges i analitzar la resposta del mòdul. Això dotaria d'un coneixement més profund del funcionament dels PIC32 per tal de poder-lo utilitzar en futurs dissenys.
- Si bé un dels objectius principals era d'alliberar de responsabilitats al mòdul, no seria una mala idea que aquest anés assumint més tasques progressivament. En vista dels resultats i del potencial del microcontrolador, aquest podria alliberar a la centraleta principal de les tasques relacionades amb l'adquisició de dades. Hi ha moltes dades que actualment arriben a la centraleta principal amb l'únic objectiu de ser enviades a la centraleta d'enregistrament de dades. La ineficiència d'aquest sistema s'exemplifica en el fet que les dades s'envien dos cops per la xarxa de CAN i això complica considerablement el cablejat. La proposta passa per connectar el mòdul a totes les xarxes de CAN per on s'enviïn dades i que aquest s'encarregui de recollir-les, tractar-les i enregistrar-les de forma mostrejada, amb la complicació que suposa la gestió del temps. Això convertiria l'actual mòdul d'enregistrament de dades en un **mòdul d'adquisició de dades**, pròpiament dit.

Agraïments

No podria acabar aquesta memòria sense dedicar unes paraules a tots aquells que m'han ajudat a fer possible aquest projecte.

En primer lloc vull fer un agraïment especial al Manuel Moreno Eguilaz, professor d'electrònica de l'escola i tutor d'aquest treball. Ell és el professor de referència del departament d'electrònica de l'equip i la seva ajuda i implicació en el projecte és un element clau per tal que l'electrònica de l'equip evolucioni any rere any. Com a tutor d'aquest treball en concret, vull agrair-li l'interès, l'ajuda i els ànims que m'ha donat al llarg de tot el curs.

També cal agrair a la direcció de l'ETSEIB l'aposta per un projecte com aquest. Cal ser conscient de l'esforç que suposa això en els moments que estem vivint. Sense aquest suport sumat a les aportacions de tots els patrocinadors, aquest projecte no podria existir.

En següent lloc, un sincer agraïment a totes aquelles persones que m'han acompanyat durant les dues temporades que he format part de l'equip ETSEIB Motorsport. Primer de tot els hi haig d'agrair la confiança que em van dipositar per entrar a formar-ne part. Ha sigut, sens dubte, la millor etapa de la carrera. M'enduc una pila de coneixements i bones experiències amb cadascun d'ells. Vull agrair en especial als companys de la secció d'electrònica: Guifré, Albert, Eulàlia i Xavi. Amb ells he après, ens hem ajudat mútuament i la bona convivència ha estat l'ingredient clau per aconseguir tot el que ens hem proposat. També agraeixo als membres d'electrònica de l'any anterior, i sobretot al Dídac, pels bons consells i l'ajut desinteressat durant tota la temporada.

I finalment, una disculpa a mode d'agraïment als més propers: parella, pares, germans, família i amics. Han sigut dos anys en els que he hagut de renunciar a moltes hores amb els que més estimo i sé que no és gens fàcil. Tot i així, ells han sigut el suport més incondicional, generós i pacient, gràcies al qual he pogut afrontar els moments més difícils.

Bibliografia

Referències bibliogràfiques

[1] GASCÓN DOMINGO, D. *Telemetria basada en WiFi per a l'equip ETSEIB Motorsport Barcelona de la Formula Student*. Barcelona, ETSEIB, 2014. [http://upcommons.upc.edu/pfc/handle/2099.1/25421, juny de 2015]

[2] KVASER. *Kvaser Leaf Light v2 User's Guide*. 2015. [http://www.kvaser.com/products/kvaser-leaf-light-v2/, juny de 2015]

[3] MAGRIÑÁ CLEMENTE, D. *Mòdul d'adquisició de dades per a l'equip ETSEIB-Motorsport Barcelona de la Formula Student*. Barcelona, ETSEIB, 2014. [http://upcommons.upc.edu/pfc/handle/2099.1/26145, juny de 2015]

[4] MICROCHIP TECHNOLOGY INC. *PIC32MX795F512H Datasheet*. 2013. [http://www.microchip.com/wwwproducts/Devices.aspx?product=PIC32MX795F512H, juny de 2015]

[5] MICROCHIP TECHNOLOGY INC. *PICkit™ 3 In-Circuit Debugger/Programmer User's Guide for MPLAB X IDE*. 2013. [http://www.microchip.com/Developmenttools/ProductDetails.aspx?PartNO=PG164130, juny de 2015]

[6] MICROCHIP TECHNOLOGY INC. *MPLAB® X IDE User's Guide v2.20*. 2014. [http://www.microchip.com/pagehandler/en-us/family/mplabx/, juny de 2015]

[7] MICROCHIP TECHNOLOGY INC. *PIC32 USB Starter Kit II User's Guide*. 2014. [http://www.microchip.com/Developmenttools/ProductDetails.aspx?PartNO=DM320003-2, juny de 2015]

[8] MICROCHIP TECHNOLOGY INC. *MPLAB® Harmony Help v1.03*. 2015. [http://www.microchip.com/pagehandler/en_us/devtools/mplabharmony/home.html, juny de 2015]

[9] MULTICOMP. *MC33191 PTC Datasheet*. 2013. [http://www.farnell.com/datasheets/1697629.pdf, juny de 2015]

[10] STMICROELECTRONICS. *RBO08 Datasheet*. 2003. [http://www.mouser.com/ds/2/389/CD00001319-250675.pdf, juny de 2015]

[11] STMICROELECTRONICS. *L7805 Voltage Regulator Datasheet*. 2014. [http://www.mouser.com/ds/2/389/CD00000444-249828.pdf, juny de 2015]

[12] TEXAS INSTRUMENTS. *SN65HVD231 3.3-V CAN Bus Transceivers Datasheet*. 2015. [http://www.mouser.com/ds/2/405/sn65hvd231-559268.pdf, juny de 2015]

[13] TEXAS INSTRUMENTS. *µA78M33 Voltage Regulator Datasheet*. 2015. [http://www.mouser.com/ds/2/405/ua78m33-555740.pdf, juny de 2015]

Bibliografia complementària

FORMULA SAE. *2015 Formula SAE® Rules*. [http://students.sae.org/cds/formulaseries/rules/, juny de 2015]

FORMULA STUDENT GERMANY. *Formula Student Electric Rules 2015*. 2015. [http://www.formulastudent.de/fse/2015/rules/, juny de 2015]

ROUELLE, C. *Optimum Race Car Engineering & Data acquisition Seminar*.

UNIVERSITAT POLITÈCNICA DE CATALUNYA. ETSEIB. DEPARTAMENT D'ELECTRÒNICA. *Apunts i pràctiques de l'assignatura de Microcontroladors*. Barcelona.